

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Sistemas Computacionales

**SISTEMA EXPERTO WEB Y MÓVIL PARA LA DETERMINACIÓN DE RUTINAS Y
DIETAS EN EL GIMNASIO FUERZA EXTREMA UTILIZANDO LA HERRAMIENTA
LIBRE CLIPS.**

Trabajo de previo a la obtención del título de Ingeniero en Sistemas
Computacionales.

Autor:

Jefferson Stalin Yacelga Almeida

Directora:

MSc. Daisy Elizabeth Imbaquingo Esparza

Ibarra, 2018



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1.- IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	1003770490
APELLIDOS Y NOMBRES:	Yacelga Almeida Jefferson Stalin
DIRECCIÓN	Joaquín Sandoval Monje 13-65 y Rafael Carvajal
EMAIL	jsyacelgaa@utn.edu.ec
TELÉFONO MÓVIL	0981438167

DATOS DE LA OBRA	
TÍTULO:	SISTEMA EXPERTO WEB Y MÓVIL PARA LA DETERMINACIÓN DE RUTINAS Y DIETAS EN EL GIMNASIO FUERZA EXTREMA UTILIZANDO LA HERRAMIENTA LIBRE CLIPS.
AUTOR(ES):	Yacelga Almeida Jefferson Stalin
FECHA:	20-11-2018
PROGRAMA:	PREGRADO

TITULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
ASESOR / DIRECTOR:	MSC. DAISY IMBAQUINGO

2.- CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 20 días del mes de noviembre de 2018.

EL AUTOR:



.....
Nombre: Jefferson Stalin Yacelga Almeida
Cedula: 1003770490



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

Ibarra, 13 de noviembre del 2018

CERTIFICACIÓN DE DIRECTOR DE TESIS

Por medio del presente yo MSc. Daisy Imbaquingo, certifico que el Sr. Jefferson Stalin Yacelga Almeida portador de la cédula de identidad Nro. 1003770490. Ha trabajado en el desarrollo del proyecto de tesis **SISTEMA EXPERTO WEB Y MÓVIL PARA LA DETERMINACIÓN DE RUTINAS Y DIETAS EN EL GIMNASIO FUERZA EXTREMA UTILIZANDO LA HERRAMIENTA LIBRE CLIPS**, previo a la obtención del título de Ingeniería en Sistemas Computacionales, lo cual ha realizado en su totalidad.

Es todo cuanto puedo certificar en honor a la verdad.

MSc. Daisy Imbaquingo

DIRECTOR DE TESIS



Ibarra, 13 de noviembre del 2018

CERTIFICACIÓN

Por medio del presente yo Ángel Núñez, certifico que el Sr. Jefferson Stalin Yacelga Almeida portador de la cédula de identidad Nro. 1003770490 quien realizó su trabajo de grado, titulado **“SISTEMA EXPERTO WEB Y MÓVIL PARA LA DETERMINACIÓN DE RUTINAS Y DIETAS EN EL GIMNASIO FUERZA EXTREMA UTILIZANDO LA HERRAMIENTA LIBRE CLIPS”**, lo ha culminado en su totalidad cumpliendo con los requisitos solicitados.

Es todo cuanto puedo certificar en honor a la verdad.


Ing. Ángel Núñez
GERENTE

GIMNASIO FUERZA EXTREMA

Dedicatoria

Dedico todo el esfuerzo realizado principalmente a DIOS, que me ha acompañado durante estos años apoyándome en cada paso que doy y dándome fuerzas para afrontar los retos que se presentan a lo largo de mi camino.

Quiero dedicar este trabajo a mis PADRES que mediante su apoyo y sacrificio he llegado a culminar una meta más en mi vida.

También lo dedico mi HERMANA que ha estado a mi lado durante los buenos y malos momentos y ha sabido apoyarme en las decisiones que he tomado.

Jefferson Yacelga

Agradecimientos

Agradezco a mis PADRES y mi FAMILIA por su apoyo incondicional y los consejos que me han brindado durante toda la vida.

Agradezco a mi directora de tesis, MSc. Daisy Imbaquingo por guiarme durante el desarrollo del presente proyecto, además le agradezco por su dedicación y compromiso que tiene como docente.

Agradezco a mis asesores de tesis, MSc. MacArthur Ortega y MSc. Marco PUSDÁ por el tiempo invertido y por guiarme en el desarrollo del presente proyecto.

Agradezco a todos mis AMIGOS que me han acompañado a conseguir este logro apoyándome y dándome fuerzas durante el camino.

Gracias por todo.

Tabla de contenido

Autorización de uso y publicación a favor de la Universidad Técnica del Norte.....	ii
Certificado del director	iv
Certificado de la empresa	v
Dedicatoria.....	vi
Agradecimientos	vii
Resumen	xiii
Abstract.....	xiv
INTRODUCCIÓN	1
Antecedentes	1
Situación actual.....	2
Prospectiva	2
Planteamiento del problema.....	3
Objetivo General	3
Objetivos Específicos.....	3
Alcance	3
Justificación	4
CAPÍTULO 1	5
MARCO TEÓRICO	5
1.1 Herramientas informáticas	5
1.1.1 Sistemas basados en conocimiento	5
1.1.2 Sistemas Expertos	7
1.1.3 Herramienta de desarrollo de sistemas expertos CLIPS	10
1.1.4 Spring Framework.....	12
1.1.5 Angular Framework.....	14
1.1.6 Ionic Framework.....	17
1.2 Base de datos PostgreSQL	21
1.3 Arquitectura de software Cliente-Servidor.....	23
1.4 Metodologías de Desarrollo	24
CAPÍTULO 2.....	30
DESARROLLO	30
2.1 Arquitectura del Sistema.....	30
2.2 Planeación del proyecto	31
2.2.1 Historias de Usuario	32
2.2.2 Características de los usuarios	37
2.2.3 Restricciones de la aplicación	37
2.2.4 Requisitos de Interfaz	37

2.2.5 Requisitos de Software	38
2.2.6 Requisitos Funcionales	38
2.2.7 Requisitos No funcionales	40
2.3 Casos de uso	41
2.3.1 Usuarios de la aplicación	41
2.3.2 Caso de uso Administrador	42
2.3.3 Casos de uso del Cliente	47
2.4 Diseño	53
2.4.1 Diseño conceptual.....	53
2.4.2 Diagrama de clases	54
2.4.3 Integración de CLIPS con Spring	54
2.4.4 Integración de Spring y Angular JS	55
2.4.5 Integración de Spring y Ionic JS.....	57
2.5 Iteraciones.....	57
2.6 Primera Iteración	58
2.6.1 Historia de Usuario - Acceso al portal web	58
2.6.2 Historia de Usuario - Registrar clientes	60
2.6.3 Historia de usuario - Administrar pago de clientes.....	61
2.6.4 Historia de usuario - Notificar fin suscripción.....	63
2.6.5 Historia de usuario - Promocionar ofertas	63
2.6.6 Historia de usuario – Reportes	64
2.7 Segunda Iteración	66
2.7.1 Historia de usuario - Aplicación móvil.....	66
2.7.2 Historia de usuario - Datos de clientes	67
2.7.3 Historia de usuario - Generar valores nutricionales	68
2.7.4 Historia de usuario - Pago pendiente del cliente.....	69
2.7.5 Historia de usuario - Registrar Rutinas del Cliente	70
2.7.6 Historia de usuario - Registrar dietas del cliente.....	70
2.8 Tercera Iteración	71
2.8.1 Historia de usuario - Registrar Ejercicios y Alimentos	72
2.8.2 Historia de usuario - Actualizar Sistema Experto.....	73
2.8.3 Historia de usuario - Determinación de rutinas y dietas del Sistema Experto	73
2.9 Pruebas.....	75
2.9.1 Pruebas Primera Iteración.....	75
2.9.2 Pruebas Segunda Iteración	78
2.9.3 Pruebas Tercera Iteración.....	81
2.9.4 Resultado de pruebas	83

CAPITULO III	87
IMPLEMENTACIÓN.....	87
3.1 Instalación	87
3.2 Requisitos.....	87
3.2.1 Requisitos del servidor para despliegue de servicios Web	88
3.2.2 Requisitos del servidor de la plataforma web del administrador	88
3.3 Alojamiento.....	88
3.3.1 Alojamiento de Spring Framework	89
3.3.2 Alojamiento de Angular Framework.....	89
3.4 Despliegue	89
CONCLUSIONES	94
RECOMENDACIONES	95
REFERENCIAS	96
ANEXOS.....	97

Índice de figuras

Figura 1. Arquitectura de los sistemas expertos.....	8
Figura 2. Ejecución de reglas en clips.....	11
Figura 3. Proceso de construcción de PhoneGap	19
Figura 4. Niveles de arquitectura cliente servidor	24
Figura 5. Diferencias entre metodologías ágiles y tradicionales	26
Figura 6. Fases de la metodología XP	29
Figura 7. Arquitectura del sistema.....	31
Figura 8. Usuarios de la aplicación	42
Figura 9. Caso de Uso Administrador	42
Figura 10. Caso de uso del Cliente	48
Figura 11. Diseño conceptual del sistema.....	53
Figura 12. Diseño de clases del sistema.....	54
Figura 13. Referencia del código para integrar CLIPS	55
Figura 14. Creación del modelo de Spring	55
Figura 15. Estructura de un proyecto con Spring	56
Figura 16. Estructura del proyecto en Angular	56
Figura 17. Interfaz de autenticación	59
Figura 18. Interfaz al acceder el usuario	60
Figura 19. Interfaz de administración de clientes	61
Figura 20. Interfaz de administración de pagos.....	62
Figura 21. Interfaz para notificaciones	63
Figura 22. Interfaz para administrar ofertas.....	64
Figura 23. Interfaz de reportes	65
Figura 24. Interfaces del aplicativo móvil.....	66
Figura 25. Interfaz de validación de clientes	67
Figura 26. Interfaces de datos del cliente en el aplicativo móvil	68
Figura 27. Interfaz de valores nutricionales.....	69
Figura 28. Interfaz de pagos en el aplicativo móvil.....	70
Figura 29. Interfaces para creación de dietas en el aplicativo móvil.....	71
Figura 30. Interfaces para la administración de ejercicios y alimentos	72
Figura 31. Interfaz de actualización del sistema experto	73
Figura 32. Creación de regla de rutinas en la base de conocimiento	74
Figura 33. Creación de regla de dietas en la base de conocimiento.....	74
Figura 34. Requisitos y casos de uso cumplidos.....	86

Índice de tablas

TABLA 1. DESCRIPCIÓN DE MÓDULOS	3
TABLA 2. HISTORIA DE USUARIO 01	32
TABLA 3. HISTORIA DE USUARIO 02.....	32
TABLA 4. HISTORIA DE USUARIO 03.....	32
TABLA 5. HISTORIA DE USUARIO 04.....	33
TABLA 6. HISTORIA DE USUARIO 05.....	33
TABLA 7. HISTORIA DE USUARIO 06.....	33
TABLA 8. HISTORIA DE USUARIO 07	34
TABLA 9. HISTORIA DE USUARIO 08.....	34
TABLA 10. HISTORIA DE USUARIO 09.....	34
TABLA 11. HISTORIA DE USUARIO 10.....	35
TABLA 12. HISTORIA DE USUARIO 11	35
TABLA 13. HISTORIA DE USUARIO 12.....	35
TABLA 14. HISTORIA DE USUARIO 13.....	36
TABLA 15. HISTORIA DE USUARIO 14.....	36
TABLA 16. HISTORIA DE USUARIO 15.....	36
TABLA 17. CARACTERÍSTICAS DEL USUARIO ADMINISTRADOR.....	37
TABLA 18. CARACTERÍSTICAS DEL USUARIO CLIENTE.....	37
TABLA 19. REQUISITOS FUNCIONALES.....	38
TABLA 20. REQUISITOS NO FUNCIONALES	41
TABLA 21. DETALLE CASO DE USO ADMINISTRADOR.....	42
TABLA 22. DETALLE CASO DE USO DEL CLIENTE.....	48
TABLA 23. DESCRIPCIÓN DE ITERACIONES	57
TABLA 24. PRUEBAS DE ACEPTACIÓN DE LA PRIMERA ITERACIÓN	75
TABLA 25. PRUEBAS DE ACEPTACIÓN DE LA SEGUNDA ITERACIÓN	78
TABLA 26. PRUEBAS DE ACEPTACIÓN DE LA TERCERA ITERACIÓN.....	81
TABLA 27. REQUISITOS Y CASOS INVOLUCRADOS DE PRUEBAS	84
TABLA 28. RESULTADOS PRIMERA ITERACIÓN	85
TABLA 29. RESULTADOS SEGUNDA ITERACIÓN.....	85
TABLA 30. RESULTADOS TERCERA ITERACIÓN	86
TABLA 31. REQUISITOS DEL SERVIDOR DE SERVICIOS WEB	88
TABLA 32. REQUISITOS DEL SERVIDOR PARA LA PLATAFORMA WEB.....	88
TABLA 33.RESULTADOS DE PRUEBAS DE ACEPTACIÓN	90

Resumen

En el gimnasio “Fuerza Extrema” de la ciudad de Ibarra, viendo la necesidad de un sistema para la administración de los clientes y en busca de un método para agilizar los resultados que obtienen las personas al asistir al gimnasio regularmente, se implementó un sistema experto que le permite al administrador tener un registro de los clientes con un control de cumplimiento de pagos y al cliente el acceso a una aplicación móvil para determinar las rutinas y dietas cumpliendo las expectativas que tiene al asistir al gimnasio y optimizando el trabajo de los entrenadores.

El proceso de desarrollo se lo realizó mediante la metodología ágil XP, basándose en las cuatro fases principales de la misma como son la planificación, diseño, desarrollo y pruebas. Esta metodología permite generar un software de calidad reduciendo el tiempo de la elaboración de documentación y se enfoca en la satisfacción del cliente incluyéndolo en el equipo de desarrollo.

Las herramientas empleadas en el desarrollo del sistema experto web móvil son: la base de datos PostgreSQL, Spring encargado de realizar las operaciones y llamadas a la base de datos, Angular para el diseño de las interfaces del administrador, Ionic para la realización de la aplicación móvil en la que acceden los clientes.

Palabras claves: Sistema Experto, Gimnasio, Programación Extrema, CLIPS, Spring, Angular, Ionic.

Abstract

In the gym “Fuerza Extrema” of the city of Ibarra, seeing the need for a system for the administration of the clients and in search of a method to accelerate the results that the people obtain when attending the gym regularly, an expert system was implemented that allows the administrator to have a record of clients with a payment compliance control and the customer access to a mobile application to determine routines and diets meeting the expectations that you have when attending the gym and optimizing the work of the coaches.

The development process was carried out using the XP agile methodology, based on the four main phases of planning, design, development and testing. This methodology allows us to generate quality software by reducing the time of documentation preparation and focuses on customer satisfaction by including it in the development team.

The tools used in the development of the mobile web expert system are the PostgreSQL database, Spring in charge of performing the operations and calls to the database, Angular for the design of the administrator interfaces, Ionic for the execution of the application mobile in which customers access.

Keywords: Expert System, Gym, Extreme Programming, CLIPS, Spring, Angular, Ionic.

INTRODUCCIÓN

Antecedentes

En la ciudad de Ibarra, provincia de Imbabura se encuentra ubicado el gimnasio “Fuerza Extrema” que fue inaugurado en el mes de agosto del 2017, contando con un espacio amplio y una gran cantidad de equipos para el acondicionamiento físico acorde a los objetivos de sus clientes y ayudándolos a mantener una dieta sana.

El gimnasio Fuerza Extrema se mantiene abierto a las nuevas tendencias, es por ello por lo que tiene acoplado varios espacios para realización de actividades como fitness, aeróbicos, cardio, pilates, etc. Además, cuenta con un sitio web en Facebook¹ para darse a conocer con sus clientes.

En los últimos años la tecnología ha aportado en varias áreas entre ellas el deporte, valiéndose de herramientas para el análisis de datos mediante la inteligencia artificial². Una rama de la inteligencia artificial son los sistemas expertos los mismos que ayudan a la mejora de toma de decisiones en las empresas ya que existe un mercado competitivo y es necesario el almacenamiento y análisis de la información.

Un sistema experto es aquel sistema de información que emula a una persona en los conocimientos que esta posee sobre un tema específico. Dicho sistema puede ser utilizado por personas que no tengan conocimiento del tema para resolver un problema e incluso puede ser utilizado por otra persona experta para guiarse mediante este sistema (Badaró, Ibañez, & Agüero, 2013).

Los sistemas expertos son capaces de procesar la información y razonar en situaciones determinadas para comunicar esa información con el usuario que pueden ser otros sistemas expertos, tomar decisiones, acorde a los requerimientos y explicar porque se ha tomado esas decisiones. También un sistema experto se lo puede tomar como un consultor para suministrar ayuda a los expertos humanos para el análisis de los datos.

Uno de los softwares más populares para realización de sistemas expertos es CLIPS, una herramienta originalmente desarrollada por el STB (Software Technology Branch), Nasa/Lyndon B. Johnson Space Center. Desde su lanzamiento en 1986, CLIPS ha sido sometido al refinamiento y mejora continua. Ahora es utilizado por miles de personas en todo el mundo. CLIPS está diseñado para facilitar el desarrollo de software para modelar el conocimiento o la experiencia humana. (Joseph C. Giarratano, 2015).

¹ Facebook es la red social más utilizada en el mundo

² Inteligencia Artificial: Programa de computación diseñado para realizar determinadas operaciones que se consideran propias de la inteligencia humana, como el autoaprendizaje

Situación actual

El gimnasio “Fuerza Extrema” se encuentra instalado en un inmueble ubicado en el sector de Alpachaca de la ciudad de Ibarra, cuenta con dos pisos, en la planta baja se realiza acondicionamiento físico con pesas y máquinas dedicadas y en la planta superior cuenta con un espacio solo para fitness o baile terapia. El gimnasio cuenta con 4 entrenadores que trabajan por turnos para guiar a los clientes en las rutinas que deben realizar cada persona y recomendar una dieta a seguir.

La mayoría de los clientes del gimnasio “Fuerza Extrema”, acuden en horas de la tarde y noche por lo que es difícil a los entrenadores un seguimiento de los clientes a todo momento en sus rutinas que deben realizar y escoger correctamente la rutina analizando sus datos como el peso y talla para obtener un ejercicio más personalizado. De igual manera los entrenadores al no poder estar todo el tiempo recopilando datos, ellos no pueden recomendar a todos sus clientes una dieta más personalizada, por lo cual algunos clientes no se muestran contentos con su servicio y dejan de asistir al gimnasio por estas razones.

El gimnasio al momento cuenta con una plantilla en Excel³ para la administración de sus clientes y los pagos que éstos deben realizar si cuentan con una suscripción y de igual manera a las personas que realizan el pago diario por lo que provoca que muchas veces los entrenadores no saben si las personas ya realizaron el pago o no obteniendo pérdidas si una persona ingresa al gimnasio sin haber pagado previamente.

Prospectiva

Mediante la implementación del sistema experto, se pretende contar con una aplicación que brinde a los clientes o al entrenador del gimnasio un sistema que permita determinar las rutinas a realizar y de igual manera la dieta apropiada por persona, dependiendo sus medidas y datos previos ingresados al sistema.

Con el conocimiento de las rutinas los clientes podrán realizar de mejor manera sus ejercicios dependiendo la zona del cuerpo que desean fortalecer, facilitando el trabajo del entrenador al poder brindar un entrenamiento mucho más personalizado por persona junto con una dieta recomendable a seguir para que los clientes logren de mejor forma sus objetivos, aumentando la eficiencia que el gimnasio consigue con sus clientes ayudado de las tecnologías que existen en la actualidad.

³ Excel es un programa informático que permite realizar tareas financieras y contables

Planteamiento del problema

¿Con la implementación de un sistema experto para la determinación de rutinas y dietas se ayuda al cliente a cumplir las expectativas al asistir al gimnasio “Fuerza Extrema” de la ciudad de Ibarra?

Objetivo General

Desarrollar un Sistema experto web y móvil para la determinación de rutinas y dietas en el Gimnasio “Fuerza Extrema” utilizando la herramienta libre CLIPS.

Objetivos Específicos

- Estudiar las herramientas informáticas, bases de datos y arquitectura de sistemas expertos para aplicarlos al proyecto propuesto.
- Desarrollar un sistema experto web móvil mediante la metodología XP
- Identificar la funcionalidad del sistema experto mediante pruebas piloto a fin de validar requerimientos de software.
- Elaborar un manual técnico del uso del sistema experto

Alcance

Desarrollar un sistema experto en un entorno web y móvil para la determinación de rutinas que puede realizar en el gimnasio y posibles dietas a seguir para el Gimnasio “Fuerza Extrema” implementando la herramienta libre Clips

El sistema contará con dos módulos principales los cuales tendrán diferentes funcionalidades que se describen en la Tabla 1.

TABLA 1. DESCRIPCIÓN DE MÓDULOS

	Gestión de Usuarios	Administración de clientes
		Datos personales de clientes Pagos de suscripción por cliente
Modulo Web (Administrador)	Gestión Rutinas y Dietas	Administración de Ejercicios disponibles Administración de valores nutricionales de alimentos Actualización de sistema experto
	Reportes	
	Datos de cliente	Administración de datos Personales Medidas y valores nutricionales Visualización de pagos

Modulo Móvil (Clientes)		Visualización de promociones
	Gestión de Rutinas	Generación de Rutinas del Sistema
		Experto
		Personalización de Rutinas
	Gestión de Dietas	Registro de rutinas
		Generación de Dietas del Sistema
		Experto
		Personalización de Dietas
		Registro de rutinas

Las herramientas principales para la gestión tecnológica que se utilizó en el proyecto se describen a continuación:

- Framework: Spring y Angular
- Lenguaje De Programación: Java, JavaScript, Html5, CSS, TypeScript
- Patrón De Diseño: MVC (Modelo, Vista, Controlador)
- Base De Datos: PostgreSQL
- Sistema Experto: CLIPS

Se manejará una arquitectura cliente servidor, en el cual del lado del cliente se utilizó Angular y consumirá los servicios de Spring para la consulta de datos y funciones del sistema experto.

Justificación

Con la falta de un entrenamiento y guía personalizada hacia los clientes del gimnasio “Fuerza Extrema” de la ciudad de Ibarra nace la idea de este proyecto con el fin de ayudar a los entrenadores y a los clientes con una base de datos donde se recopilara la información por usuario y mediante un análisis de datos poder determinar rutinas y dietas que debe realizar en el tiempo que este en el gimnasio, logrando así que el gimnasio pueda dar un seguimiento más profundo a sus clientes y a su vez los clientes estarán satisfechos con el servicio que ofrece el gimnasio.

Este proyecto pretende facilitar el trabajo de los entrenadores a la hora de escoger rutinas y dietas a seguir para sus clientes de una manera más eficaz usando tecnologías como la inteligencia artificial para determinar la mejor opción mediante la evaluación de datos del cliente permitiendo al cliente realizar sus ejercicios de manera más eficiente.

CAPÍTULO 1

MARCO TEÓRICO

1.1 Herramientas informáticas

1.1.1 Sistemas basados en conocimiento

Desde los inicios de la IA se ha tratado de que las maquinas razonen de igual manera como lo hacen los seres humanos. En este intento se creó a los Sistemas Basados en Conocimiento (SBC), con el fin de que las maquinas razonen igual que los seres humanos (Serrano, 2013).

Los SBC se componen de dos partes principales: la base de conocimiento y el motor de inferencia. En la base de conocimiento se va a guardar la información necesaria para el dominio de la aplicación, dicho de otra forma, es donde se encuentra todos los datos que permiten deducir sobre un tema específico brindados por uno o más personas con conocimiento en ese campo. La información ingresada en la base de conocimiento debe estar acorde al lenguaje de programación que utilice la misma, por lo que la información del o los expertos se la debe adaptar y darle forma antes de ser ingresada (Serrano, 2013).

El fin de los SBC no solo es guardar datos en una base de conocimiento si no que la parte más importante es que se pueda procesar estos datos y lograr que el sistema tenga su propio razonamiento mediante estos datos y por eso los SBC cuentan con un motor de inferencia que se encarga de razonar o inferir dependiendo de la información que tenga la base de conocimiento y relacionarla para conseguir una conclusión coherente con el problema que se lo haya dado (Serrano, 2013).

Los SBC ayudan a resolver problemas en donde no se puede aplicar un análisis matemático adecuado o la solución es muy compleja como por ejemplo en el área de medicina que para diagnosticar a una persona una enfermedad no se puede valer de un modelo matemático si no mediante el conocimiento de una persona que conoce el tema y el responsable para diagnosticar el mejor tratamiento.

Ventajas de utilizar SBC

Las ventajas que traen los SBC dependerán de la adecuada integración y manejo de la información. A continuación, se detallan algunas ventajas que trae consigo los SBC:

- **Respuestas consistentes:** La información de la base de conocimiento de los SBC no puede basarse en posibilidades sino más bien contiene datos de resultados anteriores o el conocimiento de diferentes expertos.

- **Disponibilidad:** Con los SBC no es necesario depender de una persona para resolver un problema ya que la información que se requiere se encuentra en la base de conocimiento y se encuentra disponible a cualquier momento
- **Preservación del conocimiento:** El conocimiento de los seres humanos es inestable ya que por cualquier motivo esta información puede ser olvidada o puede ocurrir algo con la persona que tiene este conocimiento, mientras que, si estos datos se encuentran registrados en un SBC, estos prevalecerán en el sistema por tiempo indefinido, sin el riesgo de perderlos.
- **Adquisición de nuevo conocimiento:** Ya que los datos permanecen en una base de conocimiento, permite el analizar la información que se tiene para generar una nueva a partir de este análisis.

Clasificación de los SBC

Debido a la complejidad que se encuentra al desarrollar un SBC se lo puede representar en tres formas diferentes: Sistemas Basados en Reglas, las Redes Neuronales Artificiales y el Razonamiento Basado en Casos (Cordero Morales, Ruiz Constanten, & Torres, 2013).

- **Sistemas Basados en Reglas**

Como su nombre indica estos sistemas están compuestos de reglas las mismas que deben llegar una estructura dada por un antecedente y un consecuente (Serrano, 2013). Mientras más reglas se tenga en la base de conocimiento, más precisa es la respuesta del sistema hacia un problema. Su implementación es mucha más sencilla ya que únicamente se debe tener claro todas las reglas que se va a implementar para su análisis.

- **Redes Neuronales Artificiales**

Para llenar la base de conocimiento de las redes neuronales es necesario seleccionar los ejemplos a tratar, diseñar una topología y entrenar a las redes. Las redes neuronales se basan en pesos que representan el conocimiento para llegar a una solución. Su implementación es mucho más compleja ya que requieren de un mayor análisis y cálculos para entrenar a las redes y llegar a la solución del problema. (Cordero Morales, Ruiz Constanten, & Torres, 2013)

- **Razonamiento Basado en Casos**

En este caso se trata de crear nueva información a partir de la información que se encuentra directamente en la base de conocimiento la que almacena explícitamente los datos con los que trabaja el sistema. El razonamiento basado en casos se basa

en el análisis, clasificación, interpretación, planificación de las experiencias o casos de resultados previos evaluados por el sistema (Cordero Morales, Ruiz Constanten, & Torres, 2013).

1.1.2 Sistemas Expertos

Uno de los SBC más conocidos son los Sistemas Expertos (SE) o sistemas basados en reglas. Estos sistemas tratan de simular lo que un experto con conocimiento en un área específica pueda concluir acerca de un problema planteado (Serrano, 2013). Los SE están basados en la forma en que las personas razonamos ya que mediante los conocimientos que se tiene por experiencia previa, se da una respuesta a lo que se está razonando y muchas veces se desarrolla nuevo conocimiento.

Alrededor los años 60 se desarrolla el termino Sistemas Expertos con la creación de una máquina que resolvería problemas generales llamado Generalpurpose Problem Solver que permitió abrir las puertas a la investigación y creación de sistemas expertos.

Estructura

La estructura de los SE está compuesto de dos partes, un ambiente de desarrollo diseñado para crear los componentes y llenar de información a la base de conocimiento; y la otra parte es el ambiente de consulta que es usado por el usuario final para obtener el conocimiento del SE (Badaró, Ibañez, & Agüero, 2013).

Los componentes que componen a un sistema experto son:

- **Subsistema de adquisición de conocimiento:** Este componente se enfoca en la adquisición y estructuración de la información para llenar la base de conocimiento. La transformación del conocimiento de un experto por parte del desarrollador del SE.
- **Base de conocimiento:** El conocimiento del experto se registra en esta base con el fin de comprender los datos y resolver con estos problemas. La base de conocimiento está compuesta de los elementos básicos: la heurística y las reglas que indicaran hacia donde tiene que enfocarse la información para la resolución de problemas.
- **Base de hechos:** Es la información ingresada para ser evaluada, son los hechos que se tiene sobre un problema que requiere una solución.
- **Motor de inferencia:** Es el encargado de analizar los datos del sistema, se encarga de interpretar las reglas en base a los hechos ingresados. El motor de inferencia guía a los datos hacia los pasos que debe tomar para resolver el problema.

- **Subsistema de justificación:** Si es requerido el usuario puede pedir al sistema que indique como fue el procedimiento para encontrar la solución (Badaró, Ibañez, & Agüero, 2013).

En la Figura 1 se indica la arquitectura de un sistema experto con todos los componentes detallados anteriormente.

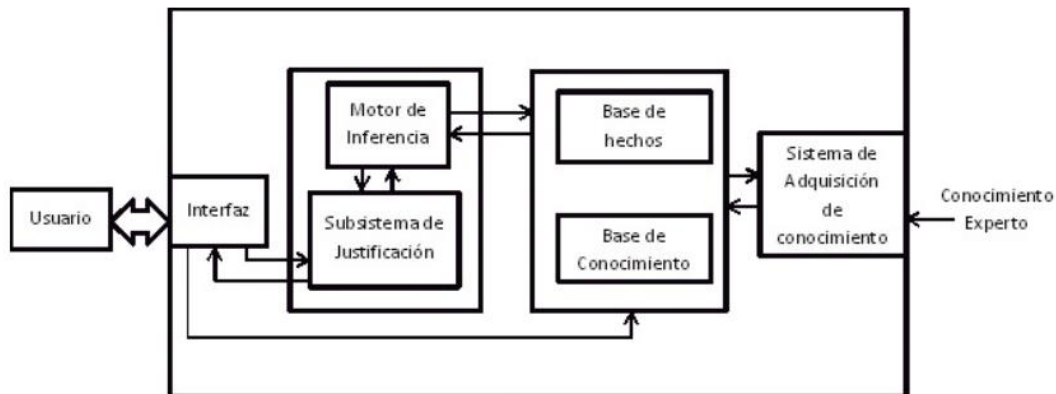


Figura 1. Arquitectura de los sistemas expertos
Fuente: (Badaró, Ibañez, & Agüero, 2013)

Funcionamiento

Se puede decir que las reglas tienen la forma de un antecedente y un consecuente, muy similar a los IF en los lenguajes de programación donde a partir de una condición o regla se llegara a hacer algo o si no cumple esta regla realizara otra acción (Badaró, Ibañez, & Agüero, 2013).

Como ejemplo en un consultorio médico para realizar el diagnostico de una persona que tiene diabetes:

- **Ej1. Si** Padece de sed Y visión borrosa **ENTONCES** paciente con diabetes tipo 1.
- **Ej2. Si** Aumenta o pierde peso Y mala sanación de heridas **ENTONCES** paciente con diabetes tipo 2.

Con la existencia de un antecedente se obtendrá un consecuente, que se encuentra guardado en la base de conocimiento o de hechos a partir de ese momento. Por ejemplo, si se cumple **Ej1** es porque hay dos hechos previos que son el padecimiento de sed y la visión borrosa. Ya que se cumple esta regla se obtiene un nuevo hecho que es la diabetes tipo 1 que se almacena en la base de hechos junto con los demás ya almacenados (Serrano, 2013).

Además de la base de hechos, los sistemas expertos realizan la búsqueda de las reglas en la base de conocimientos mediante el motor de inferencia. Para estos existen dos

estrategias que son el encadenamiento hacia adelante y el encadenamiento hacia atrás (Badaró, Ibañez, & Agüero, 2013).

Un motor de inferencia con encadenamiento hacia adelante parte de hechos va emparejando los datos de la memoria con los antecedentes que se declararon en las reglas y las va evaluando hasta que satisface el objetivo o hasta que ya no haya reglas aplicables a ese hecho. En cambio, un encadenamiento hacia atrás parte de los hechos y de un objetivo inicial y va emparejando consecuentes de las reglas con los objetivos y va aplicando las reglas hacia atrás hasta que coincidan los hechos de la memoria de trabajo (Badaró, Ibañez, & Agüero, 2013).

Herramientas

A continuación, se presenta alguna de las herramientas para desarrollo e interpretación de los SE:

- **PROLOG:** Está asociado con la inteligencia artificial, un lenguaje de programación lógica para desarrollo de SE basado en reglas. Su modelo es declarar los hechos, hacer preguntas y definir las reglas.
- **CLIPS:** Es una herramienta para el desarrollo de sistemas expertos, creada por la NASA a mediados de los años ochenta con el fin de cumplir sus objetivos y fue liberada para el desarrollo de proyectos fuera de la NASA. Su principal característica fue la adaptación con otros sistemas existentes con la utilización del lenguaje C.
- **JESS:** Esta basado en CLIPS desarrollado para el lenguaje de programación Java. Fue desarrollado en la década de los años noventa en los Sandia National Laboratories. La desventaja con JESS es que no es software libre.
- **DROOLS:** Utiliza el mismo funcionamiento en reglas que CLIPS y JESS. Consiste en una red de nodos que evalúan las entradas de la base de hechos hasta encontrar coincidencias.
- **JENA:** A diferencia de los anteriores Jena es un framework en Java con un motor de inferencia basado en normas.
- **JEOps:** Utiliza la estrategia de búsqueda mediante encadenamientos hacia adelante además de q añade normas de producción para facilitar el desarrollo de SE.
- **OpenCyc:** Esta basado en Cyc que es la base de conocimientos generales más completa con un motor de razonamiento de sentido común.

1.1.3 Herramienta de desarrollo de sistemas expertos CLIPS

Desde los inicios de los sistemas expertos, han emulado la experiencia de expertos humanos en el dominio de problemas bien definidos. El beneficio potencial de los sistemas expertos es alto: la experiencia de los expertos puede ser capturada y preservada, la enojosa tarea de que se requiera de la experiencia del humano experto una y otra vez, ya puede ser automatizada y uniformemente puede ser aplicada para la toma de decisiones.

El sistema de producción integrado en lenguaje C (CLIPS por sus siglas en inglés) es una herramienta para la construcción de sistemas expertos desarrollada por el centro espacial Johnson de la NASA, para proveer un entorno para la producción y distribución de reglas en objetos basados en sistemas expertos (Cadena, Quimí, & Peláez, 2015).

CLIPS es una herramienta con ha tenido un gran impacto para el desarrollo de SE, fue específicamente diseñado para proveer un bajo costo en la producción y pueda ser adaptado a varias plataformas. Comúnmente CLIPS ha sido usado en varias ocasiones por el sector privado. Debido que el código fuente CLIPS es fácil de entender, numerosos grupos han usado CLIPS como base para sus propios sistemas expertos. El desarrollo de la herramienta ha impulsado el desarrollo de las aplicaciones ofreciendo al sector público y privado una amplia variedad de aplicaciones y diversos entornos de desarrollo de sistemas expertos (Joseph C. Giarratano, 2015).

CLIPS fue diseñado para resolver varios problemas que tenía la NASA. Entre esos problemas estaba la posibilidad de ejecutar un sistema experto en varias plataformas, también tenían problemas para integrar estos sistemas con aplicaciones o sistemas convencionales. Como solución de estos problemas fue desarrollado CLIPS que incluso reducía costos que tenían para desarrollo y producción de sistemas expertos (Joseph C. Giarratano, 2015).

Características

CLIPS está escrito en C para su portabilidad y velocidad. Ya que este lenguaje es compatible con la mayoría de las computadoras, CLIPS puede ser instalado sin tener que modificar el código fuente de estas. Al mismo tiempo de su desarrollo, CLIPS fue una de las pocas herramientas que se escribieron en C y se adaptaron para ser ejecutadas en una amplia variedad de plataforma. Con esto CLIPS fue desarrollado para ser portable a cualquier computadora sea esta personal o de desarrollo como servidor y hardware que sea basado en lenguaje C (Joseph C. Giarratano, 2015).

Actualmente se lo puede implementar de manera gratuita para el desarrollo de sistemas expertos y ha sido constantemente actualizado y mantenido por el autor original Gary Riley

con su última versión 6.3 desarrollada en el 2015 y últimamente ha desarrollado una versión para iOS beta 0.2. Además, cuenta con un ejecutable JNI (Java Native Interface) para su integración con el lenguaje de programación Java.

Funcionamiento

El principal paradigma de programación que ofrece CLIPS es una programación basada en reglas. En este tipo de programación, las reglas son usadas para representar heurísticas⁴ cada especifica un envío de acciones para ser ejecutadas en diferentes situaciones. Una regla está compuesta de una parte “sí” y una parte “entonces”. (Joseph C. Giarratano, 2015).

La parte “sí” de una regla es una serie de patrones para que los hechos respondan a una regla que responderá en caso de que el hecho ocurra. CLIPS provee un motor de inferencias que automáticamente empareja los hechos con los patrones y determina que reglas son aplicables.

La parte “entonces” de una regla es el envío de acciones que serán ejecutadas cuando la regla sea aplicable. Las acciones que son aplicaciones para las reglas son ejecutadas cuando el motor de inferencia de CLIPS es instruido para iniciar la ejecución. El motor de inferencia selecciona una regla y luego las acciones de la selección son ejecutadas. El motor de inferencia entonces seleccionará otra regla y ejecutará otra acción hasta que ya no tenga reglas que ejecutar como se muestra en la Figura 2.

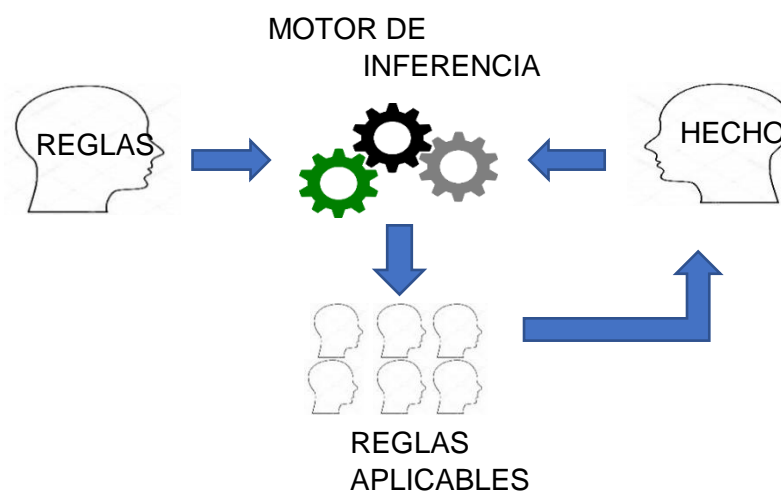


Figura 2. Ejecución de reglas en clips

⁴ Heurística: se basa en la experiencia propia del individuo, y en la de otros para encontrar la solución más viable al problema

1.1.4 Spring Framework

El desarrollo de un framework se realiza a partir de la necesidad de reutilizar código, tener una estructura del código y la reutilización de las herramientas para facilitar el trabajo del desarrollador. Un framework es definido como una herramienta o conjunto de herramientas que ayuda a los desarrolladores a trabajar de una manera más sencilla en donde podamos reutilizar el código para diferentes proyectos. Es así como en la actualidad la mayoría de las aplicaciones están basadas en algún framework para su desarrollo ya que se puede encontrar el que se adapte mejor al desarrollo de la aplicación. Tan solo teniendo un cierto modelo de proyectos o un código donde se puede insertar en varios proyectos se lo puede considerar framework ya que cumple con la reutilización de código facilitando el desarrollo.

Para la mayoría de desarrollo de frameworks populares no solo existe una persona encargada del mantenimiento y aportación con el framework, sino más bien es un grupo grande o una comunidad completa atrás de estos frameworks.

Los frameworks en el desarrollo web puede haber una infinidad entre los que ahorran en realizar secuencia de comandos en la parte del cliente y existen frameworks para la parte del servidor que es toda la lógica del aplicativo donde se conecta a base de datos o se realiza funciones, esto con la finalidad de que el lado del cliente no sea muy robusto el sistema y ahorre procesamiento de datos haciéndolo más fluido y sencillo. Para esto varios frameworks están desarrollados pensando en la seguridad, velocidad de procesamiento y la reutilización de funciones, ahorrando al desarrollador preocuparse muchas veces de estas características.

Es una época perfecta para ser un programador de Java. A través del tiempo java ha visto un poco altas y bajas, pero a pesar de algunos aspectos como los Enterprise JavaBeans (EJB), JDO (Objetos de datos de Java) y muchos sistemas de inicio de sesión, java ha pasado disfrutando de una diversa historia como plataforma sobre la que se ha creado la mayoría de software empresarial. Y un pequeño framework llamado Spring ha desarrollado un papel esencial en la historia de Java, siendo uno de las más potentes en la actualidad. (Walls, 2015)

Definición

En los inicios, Spring se creó como una alternativa a lo que son tecnologías Java empresariales que han tenido más impacto, en especial a EJB. Spring como framework ofrecía un modelo mucho más ligero que EJB (Walls, 2015).

Consecuentemente, EJB JEE (Java Enterprise Edition) evolucionaron. EJB empezó igual que Spring a ofrecer un sencillo modelo de programación propio orientado a objetos de Java. En la actualidad utiliza conceptos como dependencias (DI) y la Programación orientada a aspectos (AOP) (Walls, 2015).

Hace algunos años Spring entró en el sector de la programación con Java y el objetivo principal de simplificar el desarrollo empresarial con este lenguaje, desafiando a los grandes modelos de programación del momento con un modelo más sencillo y ligero basado en objetos de Java. (Walls, 2015)

Varios años después, con varias actualizaciones, se comprueba a que Spring ha tenido un gran impacto en el desarrollo de aplicaciones empresariales convirtiéndose en estándar para innumerables proyectos Java apoyando al desarrollo del idioma en la formación de estructuras y especificaciones que se pretendían remplazar. (Walls, 2015).

Spring Boot

Debido a la complejidad del desarrollo de aplicaciones empresariales, se creó Spring que hace posible el uso de JavaBean sencillos haciendo posible la creación de aplicaciones complejas que solo se conseguían con EJB. Sin embargo, Spring no se limita al desarrollo en el lado del servidor, gracias a su simplicidad, capacidad de prueba y acoplamiento débil; cualquier aplicación puede beneficiarse de Spring. (Walls, 2015)

Para poder reducir la complejidad que tiene Java como lenguaje para el desarrollo de aplicaciones empresariales, Spring utiliza cuatro estrategias primordiales: (Walls, 2015)

- Desarrollo ligero y mínimamente invasivo con objetos de java llamados (POJO) que permiten enfatizar el uso de clases simples sin depender de otro framework especial
- Permitir un acoplamiento débil con la inyección de dependencias y orientación de interfaces
- Programación declarativa mediante aspectos y convenciones comunes
- Reutilización de código mediante aspectos y plantillas (Walls, 2015)

Debido que Spring está inspirado en Java es posible integrar varias funciones externas que sean compatibles con Java y añadirlas a sus librerías para hacer uso de estas sin necesidad que estén instaladas en el lado del cliente. Por ejemplo, los archivos JNI (Java Native Interfaz) como el caso de CLIPS para desarrollo de sistemas expertos y tipo JAR es un tipo de archivo que permite ejecutar aplicaciones escritas en el lenguaje Java.

Aunque la ventaja principal de Spring es facilitar el desarrollo con Java existe un proyecto llamado Spring Boot que lo facilita todavía más. Spring Boot es lo más importante que ha pasado en Spring desde su creación. Añade un nuevo modelo de desarrollo a Spring y elimina el tedio que algunas veces se asocia con aplicaciones con este marco de trabajo. (Walls, 2015)

Spring Boot, en los últimos años empieza a tener un gran impacto simplificando la creación de aplicaciones y servicios, ya que normalmente las aplicaciones desarrolladas en Spring se debe configurar las conexiones del servidor o bases de datos mediante la creación de archivos escritos en xlm (*eXtensible Markup Language*, que es un metalenguaje para definir una gramática admitida para aplicaciones web) y muchas veces esto es tedioso para el programador ya que si no se define bien estas configuraciones la aplicación empieza a tener varios errores. (Walls, 2015)

Con Spring Boot se reduce el trabajo para poder desplegar la aplicación en el servidor y con esto permite enfocarse en el desarrollo de la aplicación sin preocuparse de la infraestructura ya que provee un modelo de programación parecido al lenguaje java original evitando las configuraciones XML y ofreciendo cuatro características principales para el desarrollo de aplicaciones de Spring:

- **Iniciadores de Spring Boot:** Añaden las dependencias necesarias y más comunes a una única de dependencia que se pueda añadir a la generación de un Maven o Gradle (son herramientas para la automatización de la construcción del código).
- **Configuración automática:** La función de configuración de Spring Boot recurre a la compatibilidad de configuración para determinar los vean de la aplicación y configurarlos automáticamente.
- **ILC (Interfaz de línea de comandos):** La ILC de Spring Boot se basa en el lenguaje de programación Groovy y, junto a la configuración automática, se simplifica aún más el desarrollo de aplicaciones Spring.
- **Agente:** Spring Boot Actuador añade determinadas funciones para la administración de la aplicación Spring Boot (Walls, 2015).

1.1.5 Angular Framework

El internet ha transformado completamente la forma de acceder a la información, donde millones de usuarios están a todo momento consumiendo contenido multimedia, compartiendo información, realizando transacciones, aprendiendo y accediendo a aplicaciones de entretenimiento. Actualmente, es muy poco lo que no se puede realizar con el uso del internet, que no necesiten el acceso a la web para interactuar con personas o trabajar.

Cuando se habla de aplicaciones web, muchos hacen relación a sitios populares de la web como las aplicaciones de Google, Facebook o Amazon, todas estas aplicaciones realizan transacciones complejas ya que manejan una enorme cantidad de datos para ofrecer sus servicios al público. (Solis, 2015)

Con la llegada de los dispositivos móviles y la mejora continua de los navegadores web, equipos y nuevas plataformas el desarrollo de un sitio web cada vez se hace más complejo si se quiere aprovechar todas estas funcionalidades y esto provoca que el tiempo de desarrollo sea mayor, siendo un gran problema en la industria ya que muchos emprendimientos han tenido problemas en el éxito de su producto inicial ya que no han sabido escalar su servicio a tiempo y terminan perdiendo usuarios.

Para cumplir con el planteamiento de desarrollo en menor tiempo, los programadores utilizan los frameworks para ahorrar el tiempo con la reutilización del código y otras herramientas que ya vienen definidas para el correcto funcionamiento de las aplicaciones y así cumplir con las demandas de los clientes. Una de las herramientas que ayuda al desarrollo de las aplicaciones web es Angular, un framework desarrollado por Google, una de las mayores empresas a nivel mundial. En adicional a esto Angular cuenta con una gran comunidad que da soporte a los desarrolladores con soluciones de problemas que existan. (Ginestà & Mora, 2012)

Angular actualmente es uno de los frameworks más populares y favoritos de grandes empresas y donde la mayoría de los desarrollos web apuntan actualmente. Posee grandes capacidades para el cumplimiento de la producción rápidamente y permite la escalabilidad de las aplicaciones web.

Características de AngularJS

El framework angular nos da la posibilidad de escribir una aplicación de manera sencilla, que al momento que leamos el código sea fácil de entender reduciendo la complejidad que se encuentra al momento de realizar las aplicaciones web. Angular persigue el patrón de diseño MVC, inyección de dependencias entre otras que se detallan a continuación: (Dorta, 2016)

- **Plantillas**

AngularJS está diseñado para crear aplicaciones SPA (Single Page Application) o aplicaciones de una sola página, buscando que todo el contenido y los recursos necesarios se carguen una sola vez y los va agregando, dependiendo a las acciones del usuario. Buscando una experiencia más fluida de la página con los usuarios. Este comportamiento viene acompañado por un motor de plantillas que ayuda a la generación de código dinámico con un sistema de expresiones analizadas en tiempo real.

El motor de plantillas viene con una serie de funciones para añadir más plantillas de una forma organizada y entendible. Además, permite automatizar las tareas de: iteraciones y condiciones para mostrar el contenido. Se utiliza HTML como lenguaje de plantillas y es

suficiente inteligente para detectar los cambios en la aplicación y actualizando solo lo necesario a mostrar al usuario. (Dorta, 2016)

- **Estructura MVC**

Los frameworks modernos parecidos a angular que utilizan JavaScript implementan alguna medida del patrón MVC, pero requieren separar todos los archivos en capas para agruparlos a la aplicación. Lo que nos quita tiempo en el desarrollo. Angular en cambio implementa este patrón pidiendo la separación de la aplicación en capas, pero una vez separadas el framework se encarga de lo demás. Administra los módulos creados y funciona como catalizador para unificar los elementos de la aplicación. (Dorta, 2016)

- **Vinculación de Datos**

Esta es una de las mayores y más importantes características de AngularJS, la que consiste en presentar los datos en tiempo real, uniendo varios datos. Si el valor de un dato cambia, inmediatamente el otro elemento cambia si se encuentra enlazado previamente.

La vinculación de datos es útil para crear cálculos o representaciones de gráficos en tiempo real con cambios que realice el usuario, en otros frameworks se puede utilizar esta característica añadiendo eventos y funciones adicionales, pero en AngularJS esto está integrado y no requiere ingresar más código para su implementación, solo con llamar la propiedad a los datos ya se puede crear un enlace entre estos. (Dorta, 2016)

- **Inyección de dependencias**

Desde el momento de la generación de la aplicación en Angular, está diseñada para funcionar de forma modular. La librería de Angular JS normalmente funciona con algunos elementos básicos, y si se quiere añadir nuevas funcionalidades, solo es necesario importar la librería a utilizar e inyectar la dependencia en cualquier parte del código para esté disponible, sin conflictos de instancias, ni implementaciones demasiado complejas. (Dorta, 2016)

La aplicación en AngularJS puede crecer indefinidamente creando nuevos módulos. Al usar módulos se mantiene el código encapsulado y mucho más fácil de mantener. Así logrando cada vez una mayor organización en el código y realizando buenas prácticas.

- **Directivas**

Las directivas son la forma que reconocemos cuando una aplicación está desarrollada en Angular. Las directivas en Angular JS permiten extender la sintaxis de HTML. Son elementos de programación que inyectamos en un documento web para utilizarlo en HTML, son mucho más fáciles de recordar y crean conductas para el manejo del documento.

Las directivas permiten reutilizar funciones que se encuentran ya definidas en el código de AngularJS invocando estas en donde sea necesario ahorrando código con solo incluir una etiqueta o propiedad especial en la sintaxis HTML. AngularJS también da la posibilidad de crear directivas personalizadas para extender la sintaxis HTML y hacer que la aplicación funcione de mejor manera. (Dorta, 2016)

Razones para desarrollar en AngularJS

Muchas veces el utilizar un framework nuevo en un proyecto puede salir muy costoso, porque no se conoce si todo funcionara correctamente y no se conoce el impacto que tendrá el proyecto por lo tanto se debe tener en cuenta la escalabilidad de la aplicación para que crezca con las necesidades que se presenten.

Angular es el favorito para las grandes empresas y emprendimientos por que simplifica mucho el trabajo al desarrollar aplicaciones pequeñas de un sitio web hasta aplicaciones grandes con millones de usuarios. Las características más importantes por la que AngularJS es la mejor opción al desarrollar un proyecto: (Solis, 2015)

- Es popular: Sera más fácil encontrar soporte en foros y páginas que hablen sobre el framework.
- No utiliza componentes gráficos: Esto permite al desarrollador personalizar la aplicación hasta lo más mínimo.
- Liviano y eficiente: El framework pesa 105kb y cada día lo optimizan para que no consuma muchos recursos del sistema.
- Menos Código: Con Angular se ahorrará mucho código ahorrando tiempo y desarrollando un producto de calidad con buenas prácticas.
- Compatibilidad: Angular es compatible con otros frameworks y herramientas de desarrollo web.

1.1.6 Ionic Framework

Hace unos años la mayoría de los dispositivos móviles no eran tan potentes como en la actualidad, con falta de procesamiento, sin una buena interfaz o interfaces lentas. Los primeros teléfonos inteligentes que poseían un acceso a internet solo tenían la capacidad de conectarse al correo electrónico o con un pequeño navegador podían acceder a la red, pero su experiencia era muy pobre ya que los textos eran muy simples y muchas veces no lograban cargar toda una página ya que no la soportaban. Debido a esto muchos de los que poseían estos teléfonos eran empresarios ya que la mayor utilidad que daban a estos dispositivos era el correo electrónico, mensajes de texto y llamadas. Hasta que Apple innovó la tecnología

móvil con el lanzamiento del iPhone y con el avance acelerado de la tecnología se dispone de teléfonos inteligentes muy potentes que fácilmente pueden igualar las características de hardware que un ordenador moderno. (Goins, 2017)

Con el gran impacto de los teléfonos inteligentes y su evolución, el desarrollo de aplicaciones móviles se ha convertido en una de las principales servicios que las empresas deban ofrecer a sus clientes, ya que poseen un acercamiento más personalizado con sus clientes puesto que en la actualidad los teléfonos inteligentes se han convertido en algo indispensable para la mayoría de personas, accediendo a la información de manera más rápida y sencilla, almacenando datos personales de los clientes de manera más segura y poseen una gran versatilidad en cuanto a su aplicaciones. De la misma manera que las aplicaciones de escritorio, las aplicaciones móviles realizan las mismas funciones y muchas veces son mucho más personalizadas ya que actualmente los teléfonos inteligentes poseen unas características de software cada vez más potente permitiendo desarrollar cada vez aplicaciones mucho más robustas que necesiten mayor procesamiento y así dar a los clientes un mejor servicio. (Goins, 2017)

Además, con la llegada de los teléfonos inteligentes, han surgido grandes ideas de aplicaciones para desarrollar en estos, pero algunas fueron desechadas ya que el desarrollo es muy limitado ya que se debía tener cierto conocimiento en una y cambiar toda la lógica para otra plataforma, en especial iOS contra Android. Debido a esta variedad de sistemas operativos en los dispositivos móviles, se dificultó el proceso de desarrollo y compilación de las aplicaciones móviles a diferentes plataformas.

Con la llegada de HTML 5 en el 2014, se creó una capa intermedia entre el sistema operativo nativo (Android, iOS y Windows Phone) y la aplicación desarrollada en HTML5, CSS3 y JavaScript. Esta capa se encargaría de proporcionar la vista de los recursos en forma nativa en el dispositivo móvil.

Además, surgió PhoneGap, un framework que permitió facilitar mucho más la creación de aplicaciones híbridas a los desarrolladores, con acceso a los componentes del teléfono como cámara, GPS y los diferentes sensores. PhoneGap realiza estas funciones con llamadas en el lenguaje JavaScript permitiendo acceder a funciones que solo podría realizarse de forma nativa antes de la llegada de este framework.

A continuación, en la Figura 3. se observa el funcionamiento de PhoneGap, en donde se ingresa un conjunto de elementos Html5, Css3 y JavaScript y pasa a construir la aplicación con PhoneGap para que pueda ser compatible de forma nativa para los diferentes sistemas operativos para móviles (Goins, 2017).

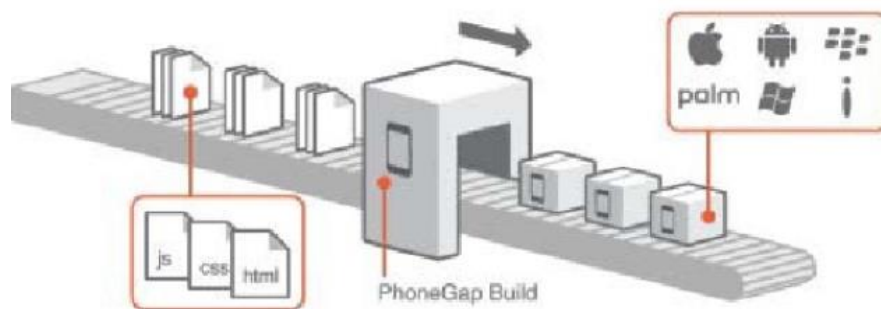


Figura 3. Proceso de construcción de PhoneGap
Fuente: (Goins, 2017)

Definición de Ionic

Para mejorar el proceso de desarrollo de las aplicaciones híbridas, en el 2013 nació Ionic Framework con la fusión de varias herramientas como PhoneGap y AngularJS. Ionic es un Kit de desarrollo de software (SDK) de código abierto que usa el concepto llamado native-feeling que quiere lograr el desarrollo de aplicaciones móviles con tecnologías web logrando que se sienta que es una aplicación nativa la que se ha desarrollado (Goins, 2017).

Ionic pretende crear aplicaciones enfocándose en la vista, proporcionando componentes para el desarrollo de la interfaz de la aplicativo, sin ser sustituto de PhoneGap que se centra en el acceso de los recursos, aunque actualmente muchas funciones de PhoneGap se las puede encontrar en Ionic Native un proyecto para poder acceder a los recursos igual que lo hace PhoneGap. En la práctica, Ionic nos ofrece una gran gama de componentes para uso de la interfaz, los controla mediante AngularJS y los compila con PhoneGap o Córdoba que se encargaran de instalar los plugin necesarios para construir la aplicación (Goins, 2017).

Características de Ionic

- **AngularJS:** Ionic utiliza la arquitectura de AngularJS basada en MVC para crear aplicaciones en una sola página optimizada para dispositivos móviles (Point, 2018)
- **Componentes CSS:** Estos componentes ofrecen un estilo personalizado, permitiendo acomodar diseños propios de la parte de la vista de la aplicación
- **Componentes JavaScript:** Ionic utiliza estos componentes para ampliar los componentes CSS con funciones de JavaScript y realizar funciones que no se las permite solo con HTML y CSS

- **Complementos de Córdoba:** Los complementos que ofrece Apache Córdoba permite transformar la aplicación desarrollada con componentes web a un lenguaje nativo con funciones JavaScript
- **Ionic CLI:** Esta es una de las utilidades que NodeJS ofrece para que mediante comandos se pueda iniciar, construir, ejecutar las aplicaciones en Ionic
- **Ionic View:** Es una plataforma que simula la aplicación en un entorno móvil para comprobar y realizar pruebas del código Ionic
- **Licencia:** Ionic esta lanzado bajo la licencia de MIT lo que significa que es software libre e impone muy pocas limitaciones para su uso (Goins, 2017)

Ventajas de Ionic

- Ionic permite el desarrollo de aplicaciones híbridas, lo que quiere decir con esto es que se las puede compilar para sistemas móviles como iOS, Android y Windows Phone, ahorrando mucho tiempo de trabajo
- Para empezar con Ionic es muy sencillo ya que para empezar Ionic ofrece una configuración predeterminada con diseños simples para su modificación y empezar a trabajar.
- Las aplicaciones están construidas por módulos lo que permite una facilidad de mantener y de actualizar la aplicación
- Ionic está respaldado por una de las mayores empresas de internet que es Google, por lo que trabajan juntos para mejorar el framework con actualizaciones y dispuestos a ayudar cuando sea necesario. (Goins, 2017)

Limitaciones de Ionic

Ionic framework igual que cualquier otro framework tiene sus limitaciones a la hora de desarrollo, a continuación, se presente algunas de estas limitaciones:

- Las pruebas de las aplicaciones iónicas pueden ser complicadas ya que se las realiza en un navegador antes de realizarlas en el teléfono y no reaccionara en algunas ocasiones de igual manera. Para realizar pruebas que utilicen funciones del teléfono, se debe construir la aplicación en un dispositivo móvil o usar un emulador.
- Existen algunas funcionalidades nativas que pueden ser difíciles de combinar, Existen algunas instancias donde se entran problemas de compatibilidad de

complementos, lo que lleva a la creación de errores difíciles de deputara. (Goins, 2017)

- Las aplicaciones híbridas usualmente son más lentas que las nativas. Sin embargo, con las nuevas actualizaciones del framework ya no sufre este problema ya que cada vez se busca la mejora del framework. (Point, 2018)
- Ya que Ionic trabaja con un navegador internamente, para su visualización, puede tener menos funcionalidades que normalmente tendría una aplicación nativa desarrollada directamente para el sistema operativo del teléfono, debido a que no trabaja directamente con el sistema.

1.2 Base de datos PostgreSQL

PostgreSQL es un gestor de base de datos orientado a objetos usado para varios entornos de software libre y conocido por sus funcionalidades avanzadas que soporta como la creación de funciones agregadas y la utilización de estas en construcciones de ventana, tablas comunes y expresiones de tablas recursivas, además incluye una funcionalidad de replicación de transmisión. Esta característica de PostgreSQL raramente se encuentra en bases de datos open source por lo que se encuentra al mismo nivel de muchos SGBD (Sistemas de gestión de base de datos) comerciales. (Regina & Leo, 2015)

Además de tener unas características avanzadas como software comercial. También es mucho más rápido, superando a muchas otras bases de datos, incluyendo a las de software propietario y muchos tipos de cargas de trabajo de bases de datos. (Regina & Leo, 2015)

PostgreSQL se distribuye bajo licencia BSD (*Berkeley Software Distribution*), lo que permite el uso, distribución y modificación, con la única restricción de que mantenga el copyright del software a sus autores la Universidad de California y al grupo de PostgreSQL Global Development Group. La ventaja que tiene PostgreSQL es que está disponible para múltiples plataformas en especial las modernas basadas en Unix y Windows de forma nativa. (Ginestà & Mora, 2012)

El desarrollo de PostgreSQL se remonta a inicios de 1986 en la Universidad Berkeley de California, que tenía de nombre POSTGRES. En el diseño y desarrollo del software se incluyeron los conceptos de base de datos avanzados y el soporte a la orientación de objetos.

Ilustra fue la empresa que comercializaba POSTGRES donde posteriormente fue a formar parte de la empresa Informix. A los investigadores y académicos les tomaba mucho tiempo mantener el proyecto, por lo que en 1993 se liberó la versión 4.5 y se dio por terminado el proyecto. En 1994, se le incluyó SQL en POSTGRES para posteriormente liberar el código

con el nombre de Postgres95 con múltiples cambios al código original mejorando el rendimiento y para su mayor entendimiento. El nombre PostgreSQL que actualmente se lo conoce se lo dio en 1996 con la liberación de la versión 6.0. Y actualmente PostgreSQL está en la versión 10.1 y tiene una versión estable 9.6.6. (Ginestà & Mora, 2012)

Lo que hace especial a PostgreSQL es que no es solo una base de datos porque también es una plataforma de aplicaciones que permite escribir procedimientos almacenados y funciones en algunos lenguajes de programación. Adema cuenta con el soporte para más idiomas con el uso de extensiones. Por ejemplo, se puede escribir funciones almacenadas en SQL y PL/pgSQL, lenguajes con extensiones para PL/Python, PL/Perl, PL/JavaScript, son algunos de las distribuciones más comunes. También tiene la posibilidad de definir funciones agregadas que contengan solo SQL. A diferencia de MySQL y SQL Server, no requiere compilación para construir una función agregada en PostgreSQL. (Regina & Leo, 2015)

El soporte personalizado de PostgreSQL es fácil de usar y muy sofisticado, rivalizando y superando a muchas bases de datos relacionadas. Oracle es el competidor más cercano en términos de soporte personalizado. Se puede definir nuevos tipos de datos en PostgreSQL con una matriz complementaria para usar como un tipo de columna de tabla. Además de tener la capacidad de definir nuevos tipos de datos, también se puede definir funciones, operadores y enlaces de índice para trabajar con estos tipos. Muchas de las características de las extensiones de terceros se las aprovecha para mejorar el rendimiento, proporcionar mejor construcción del código obteniendo un código más corto y fácil de mantener, y se puede realizar tareas que en muchas bases de datos es imposible realizarlas. (Regina & Leo, 2015)

Características principales

Las características que nos ofrece PostgreSQL, que lo hace competir con la mayoría de base de datos relacionales son:

- Esta desarrollado en C mediante la utilización de herramienta como Yacc y Lex
- La API de acceso al SGBD de PostgreSQL se encuentra disponible en la mayoría de los lenguajes de programación
- Cuenta con un extenso conjunto de tipo de tipo de datos, permitiendo su extensión mediante tipos y operadores definidos por el usuario
- La administración de PostgreSQL se basa en usuarios y privilegios de estos
- Se puede conectar mediante TCP/IP y sockets, además de soportar completamente el acceso a la base de datos estándar ODBC (Open Database Connectivity)
- Tiene soporte para idioma español, y los mensajes de error en este idioma
- Es altamente confiable cuanto a estabilidad

- Puede añadirse a librerías externas para soportar la encriptación.
- Mejora las operaciones de bloqueo y transacciones en sistemas multiusuario
- Soporta vistas, integridad referencial, procedimientos almacenados, subconsultas y la mayoría de los tipos y operadores soportados por otras bases de datos
- En PostgreSQL se puede definir un nuevo tipo de tabla debido a su implementación de algunas extensiones de orientación a objetos. (Ginestà & Mora, 2012)

Limitaciones

Se puede encontrar algunas limitaciones en PostgreSQL, siendo tres las más importantes que son:

- Puntos de recuperación dentro de transacciones ya que si existe un fallo mientras se ejecuta la transacción, esta aborta completamente.
- No se puede definir donde almacenar la base de datos, esquemas, índices, etc.
- El soporte a orientación a objetos no es un soporte completo ya que solo es una extensión (Ginestà & Mora, 2012).

1.3 Arquitectura de software Cliente-Servidor

La popularidad de internet ha ido creciendo a grandes pasos y con la evolución de la tecnología, se aprovecha mucho los recursos como mejoras de microprocesador, comunicaciones ópticas y capacidad de almacenamiento, lo que ha traído consigo que el desarrollo de aplicaciones web se acreciente valiéndose de estas nuevas tecnologías y aumentando la satisfacción del cliente ya que para ejecutar una aplicación web solo se requiere de un navegador y un dispositivo con acceso a internet (Deitel, Deitel, & Deitel, 2014).

Con frecuencia las aplicaciones web son de multiniveles también conocidas como aplicaciones de n niveles para dividir las funcionalidades en n niveles. Los niveles pueden estar en una misma computadora, pero lo común es que estos niveles estén en varias computadoras para repartir el procesamiento. La mayoría de las aplicaciones web cuentan de tres niveles como muestra la Figura 4. donde podemos observar que el nivel superior es la parte del cliente que es toda la parte grafica de la aplicación como botes, formularios, imágenes, etc.; un nivel intermedio donde está el servidor que se encarga de la lógica del sistema y se realiza operaciones que solicite el cliente; y por último el nivel inferior al de base de datos donde se encuentra toda la información del sistema (Deitel, Deitel, & Deitel, 2014).



Figura 4. Niveles de arquitectura cliente servidor
Fuente: (Deitel, Deitel, & Deitel, 2014)

Los comandos en el lado del cliente tienen limitaciones, como la dependencia de navegadores: en cuanto al hardware y del sistema de archivos local en cuanto a la seguridad. Si se realiza toda la programación en el lado del cliente, el mismo tiene acceso a todas las secuencias de comandos si la herramienta de ver el código fuente que muchos navegadores lo traen. La información de la base de datos, como datos personales de los usuarios o contraseñas, no debe estar en el cliente. Toda la validación de datos se tiene que reflejar en el lado del servidor (Deitel, Deitel, & Deitel, 2014).

Los desarrolladores tienen más flexibilidad con comandos del lado del servidor, que envían respuestas más personalizadas hacia los clientes. Los datos que reciben del servidor es información actualizada o en tiempo real de una base de datos, como, por ejemplo, un cliente podría conectarse al servidor web de una aerolínea y solicitar los vuelos de Quito a Guayaquil entre el 20 de abril y 16 de mayo. El servidor se encarga de consultar los datos a la base de datos para generar un documento JSON con todos los datos de los vuelos y enviar ese documento al cliente, el que se encargaría de presentar la información al usuario final mediante una interfaz gráfica.

Los lenguajes de secuencias de comandos del lado del cliente tienen un rango más amplio en cuanto a las capacidades programables que el lado del cliente puede ofrecer. Las secuencias de comandos en el servidor también tienen acceso a software en el servidor para extender la funcionalidad de este (Deitel, Deitel, & Deitel, 2014).

1.4 Metodologías de Desarrollo

A lo largo de la historia del software, se han desarrollado millones de proyectos y muchos más por desarrollar. Pero se ha notado un patrón en cuanto a los proyectos que han tenido mayor éxito, el cual es la estructuración y planificación de proyectos con metodologías de desarrollo. Debido al éxito que han tenido estos proyectos por el uso de las metodologías de desarrollo, se fue añadiendo mayor interés en estas metodologías.

Metodologías Ágiles

Las metodologías de desarrollo tradicionales se centraban en el proceso que deben realizar, las actividades que deben cumplir para llevar a cabo el proceso y las herramientas que usaran. Estas metodologías tuvieron problemas en algunos proyectos que las implementaron, por lo que se establecieron algunas mejoras como más procesos, más artefactos y restricciones; lo que provocó que el proceso de desarrollo vaya haciéndose mucho más complejo aumentando costos a la empresa que los desarrollan. Estas metodologías funcionan correctamente cuando el proyecto es muy grande. Pero actualmente las empresas prefieren estimar gastos en el presupuesto y desean que el proyecto se entregue en el menor tiempo posible (Canós, Letelier, & Carmen, 2012).

Debido a los inconvenientes que traen las metodologías tradicionales, se presentaron las metodologías ágiles que brindan una mayor colaboración con el cliente y un desarrollo incremental del software. Se ha demostrado que estas metodologías son eficientes cuando los proyectos son muy cambiantes además q reducen los tiempos de desarrollo, manteniendo la misma calidad que las metodologías tradicionales.

Comparación de las metodologías tradiciones y metodologías ágiles

En febrero de 2001 junto a un grupo de 17 expertos de la industria de software, nace el término “ágil” aplicado al desarrollo de software. Con las metodologías ágiles se pretende crear valores y principios para los equipos de desarrollo en cuanto la realización de un proyecto de software de calidad en menor tiempo y mantenerse en contacto con el cliente para realizar los cambios durante el desarrollo facilitando el proceso y ofreciendo una alternativa a las metodologías tradicionales (Canós, Letelier, & Carmen, 2012). En la Figura 5 se presenta un cuadro con las diferentes diferencias que tiene una metodología con la otra:

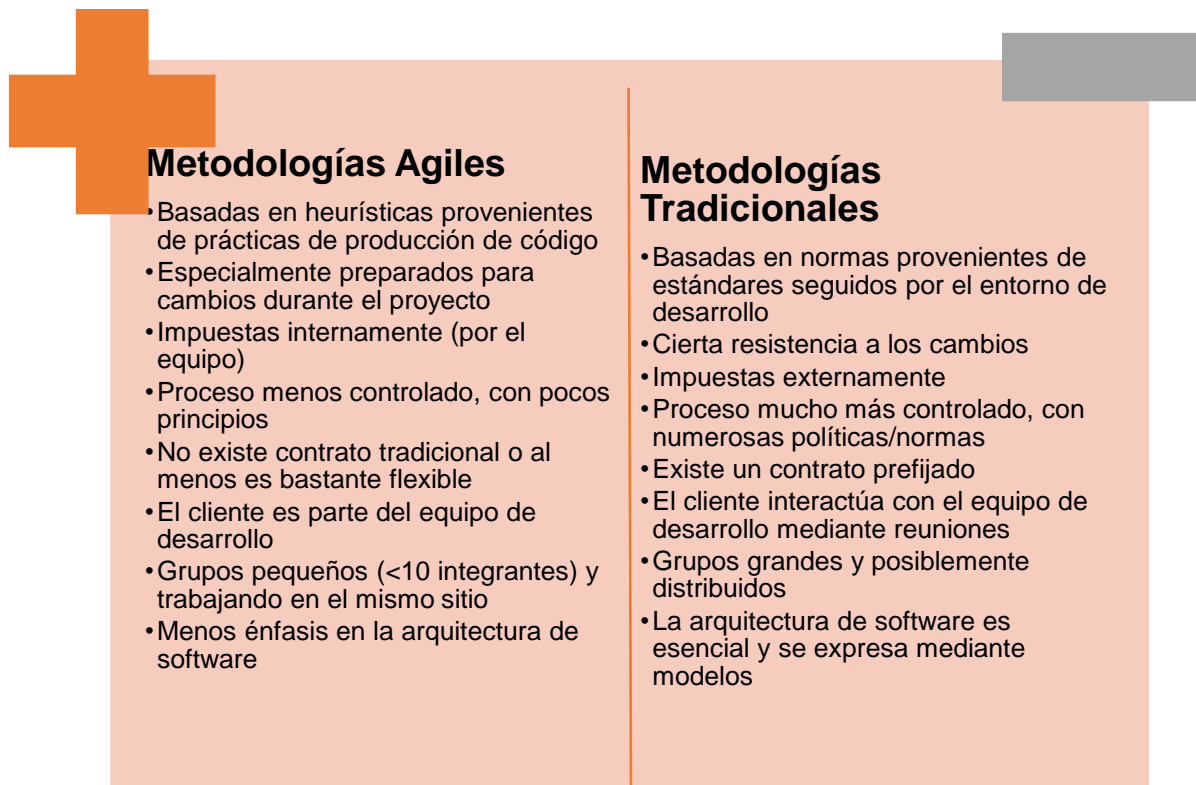


Figura 5. Diferencias entre metodologías ágiles y tradicionales

Características para implementar metodología ágil

Con las metodologías ágiles se quiere llegar a mejorar la producción de un producto y de igual forma su calidad ya que se encuentran abiertas a cambios y dando mayor satisfacción al cliente entregando el producto en menor tiempo. La implementación de las metodologías ágiles trae consigo varias características que se beneficiara los clientes como:

- Procesos y definiciones de productos más flexibles: Con fácil adaptación a los cambios tecnológicos a favor del cliente
- Retroalimentación continua: Se trabaja juntamente con el cliente presentando entregables y respondiendo a los cambios del cliente con mayor rapidez
- Mejorar la calidad: Ya que las metodologías ágiles se basan en metodologías tradicionales que se enfocan en la calidad del producto, el objetivo es el mismo entregar un producto de calidad.
- Aclaran proyectos que no están bien definidos: Debido al trabajo junto al cliente, se puede ir clarificando los requisitos mediante el proyecto avanza

Metodología ágil XP

La programación extrema (XP), es el enfoque más utilizado por el desarrollo de software ágil. Los principales signos e ideas para desarrollar esta metodología ocurrieron en la década de 1980, el principal autor de esta metodología fue Kent Beck, el mismo que estuvo en la primera prueba piloto que se realizó a la metodología en 1996. Se la denominó metodología Extrema por el hecho que lleva al extremo las buenas prácticas de la ingeniería de software. (Laínez Fuentes, 2015)

La metodología XP tiene como base que exista continuamente la comunicación del cliente y el equipo de desarrollo, además pretende ofrecer soluciones más simples al momento de implementación y permite la realización de cambios durante el desarrollo del proyecto para entregar un software de calidad al cliente.

XP se encuentra dirigida a proyectos pequeños con equipos de trabajo de 2 a 12 personas. Pero algunos afirman que puede ser utilizada en proyectos grandes, dividiendo en pequeños módulos o subproyectos independientes. Los proyectos deben cumplir las siguientes características, por lo menos algunas de ellas para que sea eficiente el uso de la metodología. (Laínez Fuentes, 2015)

- Cuando no se tiene una idea bien definida y los requisitos cambian con frecuencia
- Desarrollo de sistemas orientados a objetos
- Equipos pequeños
- Desarrollo incremental

Valores de la metodología XP

La metodología XP se encuentra definida por 5 valores que propone para que describir los objetivos de largo plazo y de estos dependerá el éxito que tendrá el proyecto (Laínez Fuentes, 2015). Los valores son la comunicación, la simplicidad, retroalimentación, respeto y coraje.

- **La comunicación:** XP se basa todo en el equipo de trabajo, es importante tener una comunicación clara y fluida para que el equipo de trabajo encuentre soluciones a los problemas que se presenten.
- **La simplicidad:** Es importante que el desarrollo sea lo más simple posible, desarrollando solo lo necesario para no mal gastar el tiempo en detalles que no son requeridos.
- **Retroalimentación:** El objetivo de XP es entregar lo requerido en el menor tiempo posible a cambio que el cliente realice una retroalimentación con el fin de conocer sus nuevos requerimientos y realizarlos lo más pronto posible.

- **Respeto:** Se mantiene el respeto entre el cliente y los desarrolladores.
- **Coraje:** El equipo de desarrollo debe tener el valor para decir la verdad sobre el avance del proyecto, evitando excusas para esconder errores. (Bahit, 2012)

Fases de metodología XP

Las fases principales de la metodología XP son planeación, diseño, codificación y pruebas.

- **Planeación:** Esta fase comienza escuchando. Se recaba los requerimientos para el desarrollo para que los miembros del equipo entiendan el propósito del proyecto y recabar características y funciones principales que requiere el cliente. Mediante la escucha se describen lo que la aplicación debe realizar o mostrar, características y funcionalidad que tendrá el software; estas son las historias del usuario que serán escritas por el cliente. Posteriormente, se evalúan las historias para dar un valor o costo medido en semanas de desarrollo. En el proceso de desarrollo se puede generar nuevas historias. En la fase de planeación se realiza un plan para el desarrollo. (Romero & Elena, 2015)
- **Diseño:** XP basa su diseño en el principio MS (mantenlo sencillo). Los diseños sencillos siempre son preferidos sobre los más complejos. Mediante la utilización de tarjetas CRC (clase-responsabilidad-colaborador) XP pretende desarrollar el software en un contexto orientado a objetos. Estas tarjetas identifican y organizan las clases orientadas a objetos. Si en el diseño las historias encuentran un problema para el diseño, XP recomienda crear un prototipo operativo de esa funcionalidad del diseño. Se crea un prototipo del diseño llamado solución en punta, con el objetivo de disminuir el riesgo cuando ya se empieza el desarrollo e implementación original de la historia. Una de las características principales del diseño en XP es que este ocurre antes como después de que comienza la codificación (Romero & Elena, 2015).
- **Codificación:** La tercera fase de XP es desarrollar una serie de pruebas unitarias para cada historia que se van a incluir en la entrega. Con la creación de la prueba el desarrollador está capacitado para enfocarse en lo que se deberá implementar para pasar la prueba. Cuando el código está terminado, se aplica una prueba unitaria, para tener una retroalimentación hacia los desarrolladores (Romero & Elena, 2015).
- **Pruebas:** Las pruebas unitarias que se crean deben implementarse en una estructura que permita automatizarlas. Esto es cuando se modifique el código se pueda correr estas pruebas en repetidas ocasiones y fácilmente. Además, se añaden unas pruebas de aceptación XP, son las pruebas que realiza el cliente y se centran en las

características generales del sistema que se puedan revisar y probar por el cliente (Romero & Elena, 2015).

Las fases de la metodología XP sirven de guía para que el equipo de desarrollo tenga en cuenta el ciclo que debe cumplir la metodología con la retroalimentación del cliente. En la Figura 6 se resume el ciclo de las fases de la metodología:



Figura 6. Fases de la metodología XP

CAPÍTULO 2

DESARROLLO

2.1 Arquitectura del Sistema

El proyecto está basado en una arquitectura cliente-servidor que brinda un desarrollo más organizado, separando las responsabilidades entre la lógica de la aplicación y lo que el cliente consume, con esto se tiene una idea más clara de cómo está estructurada la aplicación y es mucho más sencillo su mantenimiento.

El cliente usualmente es el que tiene contacto con los usuarios finales mediante una interfaz, el cliente inicia solicitudes al servidor actuando como un remitente en la aplicación y se mantiene a la espera para recibir las respuestas del servidor que normalmente son los datos de una aplicación. En este caso existen los clientes son una aplicación web creada con el framework AngularJS y una aplicación móvil con el framework Ionic, ambos frameworks contiene código HTML5, CSS3, JavaScript y TypeScript.

El servidor es el receptor de las peticiones del cliente, en el servidor se encuentra toda la lógica del sistema, se encarga de recibir las solicitudes que lleguen de los clientes y procesarlas para devolver una respuesta a los clientes. Para la creación del servidor en este proyecto se lo realizo con el framework Spring que contiene el código en Java y el gestor de base de datos PostgreSQL.

En una arquitectura cliente-servidor, no es necesario tener alojado los dos en una misma máquina, lo que permite que el servidor se encargue solamente en llevar a cabo las funciones que el cliente solicite, haciendo con esto que la carga del cliente sea más ligera y mucho más rápida. En la Figura 7 se detalla mediante gráficos la arquitectura que tiene el sistema experto web móvil y las herramientas que se utilizó.

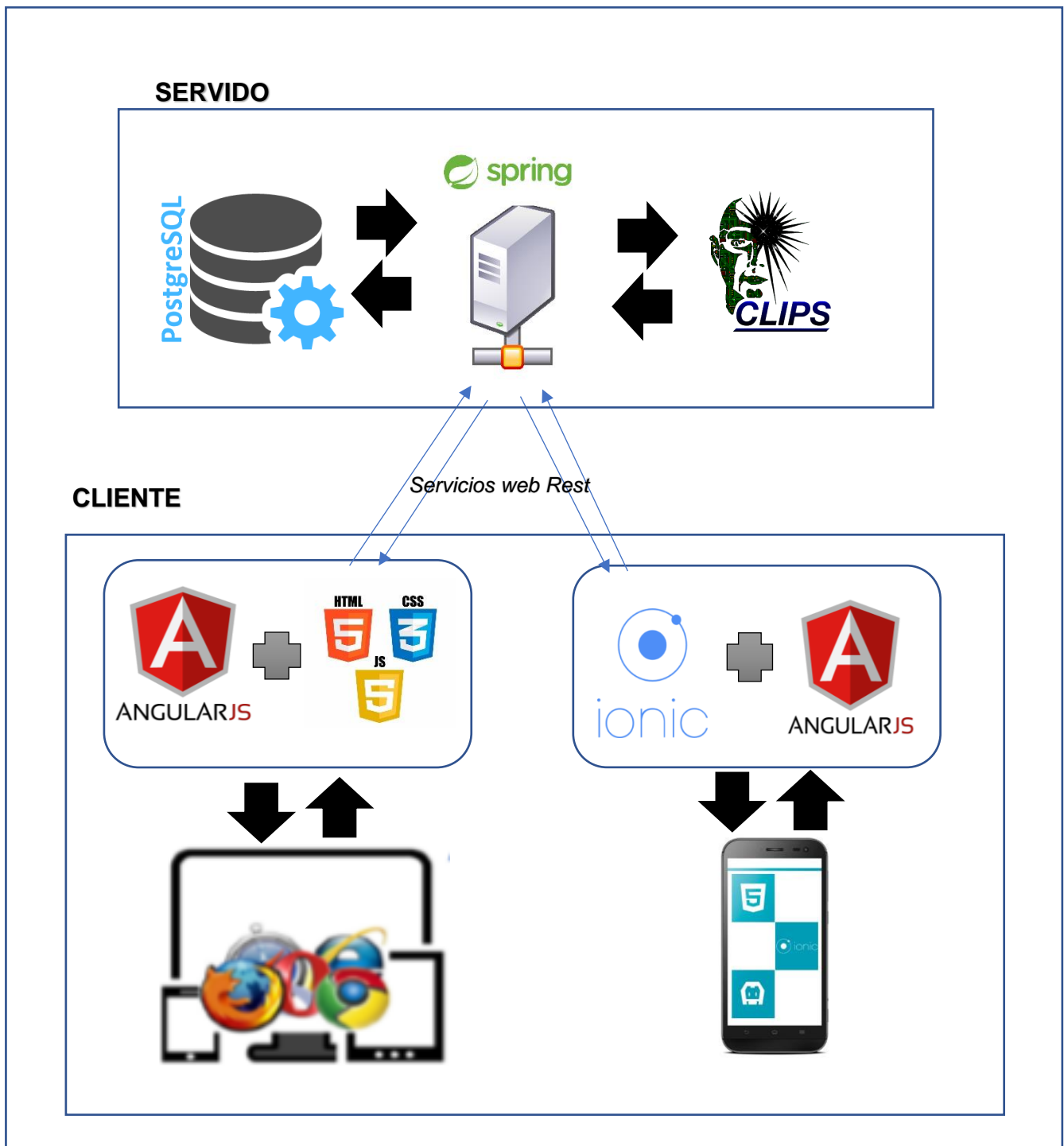


Figura 7. Arquitectura del sistema

2.2 Planeación del proyecto

Conforme a la metodología XP, es muy importante tener comunicación constante con el cliente mediante entrevistas y usando un lenguaje simple para obtener la mayor información necesaria para la entrega del producto final. En esta fase de planificación se la realiza con el objetivo de recopilar datos de la aplicación para generar historias de usuario y estimar tiempos en el desarrollo de la aplicación.

2.2.1 Historias de Usuario

Las historias de usuario que se recopilaron con el fin de detallar los requisitos que debe cumplir el sistema, se las generaron en las distintas reuniones con los dirigentes del Gimnasio tomando en cuenta sus requisitos para la formación de la aplicación.

TABLA 2. HISTORIA DE USUARIO 01

Historia de Usuario	
Número: 01	Usuario: Administrador
Nombre historia: Acceso al Portal Web	
Prioridad: Alta	Riesgo en desarrollo: Bajo
Programador: Sr. Jefferson Yacelga	Iteración asignada: 1
Descripción: El portal web estará disponible para el administrador desde cualquier navegador. Al cual podrá acceder mediante un usuario y una contraseña.	

TABLA 3. HISTORIA DE USUARIO 02

Historia de Usuario	
Número: 02	Usuario: Administrador
Nombre historia: Registrar Clientes	
Prioridad: Media	Riesgo en desarrollo: Bajo
Programador: Sr. Jefferson Yacelga	Iteración asignada: 1
Descripción: El administrador creará nuevos usuarios para el acceso a la aplicación móvil y también podrá dar de baja a los clientes ya registrados.	

TABLA 4. HISTORIA DE USUARIO 03

Historia de Usuario	
Número: 03	Usuario: Administrador
Nombre historia: Administrar pago de clientes	
Prioridad: Media	Riesgo en desarrollo: Medio
Programador: Sr. Jefferson Yacelga	Iteración asignada: 1
Descripción: El administrador tendrá un registro de los pagos realizados por los clientes, los pagos pueden ser los puede realizar en cuotas.	

TABLA 5. HISTORIA DE USUARIO 04

Historia de Usuario	
Número: 04	Usuario: Administrador
Nombre historia: Notificar Fin Suscripción	
Prioridad: Media	Riesgo en desarrollo: Medio
Programador: Sr. Jefferson Yacelga	Iteración asignada: 1
Descripción: El sistema notificara por día, los clientes que ya finalizaron su suscripción o están próximos a finalizar, para recordar dar de baja a los clientes.	

TABLA 6. HISTORIA DE USUARIO 05

Historia de Usuario	
Número: 05	Usuario: Administrador
Nombre historia: Promocionar ofertas	
Prioridad: Media	Riesgo en desarrollo: Medio
Programador: Sr. Jefferson Yacelga	Iteración asignada: 1
Descripción: El sistema presentara ofertas en la aplicación de los clientes para que ellos puedan visualizar.	

TABLA 7. HISTORIA DE USUARIO 06

Historia de Usuario	
Número: 06	Usuario: Administrador
Nombre historia: Reportes de Clientes	
Prioridad: Media	Riesgo en desarrollo: Bajo
Programador: Sr. Jefferson Yacelga	Iteración asignada: 1
Descripción: El administrador podrá generar reportes de todos los clientes registrados y los clientes que estén dados de baja por el administrador.	

TABLA 8. HISTORIA DE USUARIO 07

Historia de Usuario	
Número: 07	Usuario: Clientes
Nombre historia: Aplicación móvil	
Prioridad: Baja	Riesgo en desarrollo: Medio
Programador: Sr. Jefferson Yacelga	Iteración asignada: 2
Descripción: Los clientes podrán acceder a una aplicación móvil del gimnasio mediante su correo electrónico y una contraseña.	

TABLA 9. HISTORIA DE USUARIO 08

Historia de Usuario	
Número: 08	Usuario: Clientes
Nombre historia: Datos de Clientes	
Prioridad: Media	Riesgo en desarrollo: Medio
Programador: Sr. Jefferson Yacelga	Iteración asignada: 2
Descripción: Los clientes del gimnasio registrarán una contraseña para acceso de la aplicación, además podrán actualizar sus datos personales y datos variables como el peso, altura, edad y medidas; para el funcionamiento del Sistema Experto	

TABLA 10. HISTORIA DE USUARIO 09

Historia de Usuario	
Número: 09	Usuario: Administrador
Nombre historia: Generar valores nutricionales	
Prioridad: Alta	Riesgo en desarrollo: Alto
Programador: Sr. Jefferson Yacelga	Iteración asignada: 2
Descripción: El cliente podrá visualizar sus valores nutricionales como IMC, peso ideal, calorías diarias necesarias, proteínas, carbohidratos y grasas recomendadas.	

TABLA 11. HISTORIA DE USUARIO 10

Historia de Usuario	
Número: 10	Usuario: Cliente
Nombre historia: Pago pendiente del cliente	
Prioridad: Media	Riesgo en desarrollo: Medio
Programador: Sr. Jefferson Yacelga	Iteración asignada: 2
Descripción: El cliente podrá visualizar el último pago realizado y tendrá un registro de los pagos realizados.	

TABLA 12. HISTORIA DE USUARIO 11

Historia de Usuario	
Número: 11	Usuario: Administrador
Nombre historia: Registrar Rutinas del Cliente	
Prioridad: Media	Riesgo en desarrollo: Medio
Programador: Sr. Jefferson Yacelga	Iteración asignada: 2
Descripción: El cliente podrá generar su propia rutina diaria basándose en los ejercicios que estén registrados en el sistema. Además, tendrá un registro de las rutinas realizadas en los días anteriores.	

TABLA 13. HISTORIA DE USUARIO 12

Historia de Usuario	
Número: 12	Usuario: Administrador
Nombre historia: Registrar Dietas del Cliente	
Prioridad: Media	Riesgo en desarrollo: Alto
Programador: Sr. Jefferson Yacelga	Iteración asignada: 2
Descripción: El cliente podrá generar su propia dieta diaria teniendo en cuenta las calorías, proteínas, carbohidratos y grasas que el sistema le recomienda consumir; como ayuda el cliente se podrá basar en los alimentos y sus valores nutricionales que registre el administrador. Además, tendrá un registro de las dietas realizadas en los días anteriores.	

TABLA 14. HISTORIA DE USUARIO 13

Historia de Usuario	
Número: 13	Usuario: Administrador
Nombre historia: Registrar Ejercicios y Alimentos	
Prioridad: Media	Riesgo en desarrollo: Bajo
Programador: Sr. Jefferson Yacelga	Iteración asignada: 3
Descripción: El administrador creará los ejercicios y alimentos con sus valores nutricionales, para ser usados como complemento del Sistema Experto	

TABLA 15. HISTORIA DE USUARIO 14

Historia de Usuario	
Número: 14	Usuario: Administrador
Nombre historia: Actualizar Sistema Experto	
Prioridad: Alta	Riesgo en desarrollo: Alto
Programador: Sr. Jefferson Yacelga	Iteración asignada: 3
Descripción: El administrador podrá descargar la base de conocimiento para poder realizar actualizaciones y subir nuevamente al servidor	

TABLA 16. HISTORIA DE USUARIO 15

Historia de Usuario	
Número: 15	Usuario: Clientes
Nombre historia: Determinación de rutinas y dietas del Sistema Experto	
Prioridad: Alta	Riesgo en desarrollo: Alto
Programador: Sr. Jefferson Yacelga	Iteración asignada: 3
Descripción: Los clientes podrán seleccionar que tipo de ejercicio desean realizar para que el sistema genere una rutina tomando en cuenta sus datos ingresados. De la misma manera la dieta deberá generarse automáticamente, tomando en cuenta los valores nutricionales	

2.2.2 Características de los usuarios

Las principales características de los usuarios del sistema experto web móvil se detallan a continuación:

TABLA 17. CARACTERÍSTICAS DEL USUARIO ADMINISTRADOR

Tipo de Usuario	Administrador del Sistema
Formación	Ingeniero en Sistemas Computacionales
Habilidades	Conocimiento y experiencia en Acondicionamiento Físico, y nutrición
Actividades	Administrar clientes, pagos y datos para el funcionamiento del sistema experto

TABLA 18. CARACTERÍSTICAS DEL USUARIO CLIENTE

Tipo de Usuario	Cliente (Usuario de la App Móvil)
Formación	Desconocida
Habilidades	Utilizar aplicaciones móviles
Actividades	Registrar datos personales, generar rutinas y dietas personales;

2.2.3 Restricciones de la aplicación

El sistema esta creado para un gimnasio de la ciudad de Ibarra, que abrió sus puertas en el 2017 y tiene poco tiempo de apertura, por lo que no cuenta con los recursos necesarios para pagar costos de licencia de software o herramientas muy costosas. Por lo que se limitó al uso de software libre como es el uso de la base de datos PostgreSQL, para la realización del sistema experto se escogió CLIPS y además de los frameworks Angular, Spring y Ionic.

2.2.4 Requisitos de Interfaz

El sistema tendrá dos interfaces principales, uno para el usuario administrador que tendrá un entorno web accesible desde cualquier navegador y una aplicación móvil para el uso de los clientes del gimnasio en donde se hará uso del sistema experto.

2.2.5 Requisitos de Software

Las herramientas para el desarrollo del sistema web y móvil son de distribución libre, de igual manera la base de datos es de licencia libre. Los servidores son de paga ya que están en Internet.

2.2.6 Requisitos Funcionales

Los requisitos funcionales derivados del análisis de las historias se presentan en la Tabla 19:

TABLA 19. REQUISITOS FUNCIONALES

Número de requisito	RF-01
Nombre de requisito	Acceso al Portal Web
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requisito	El acceso al portal web lo realiza únicamente el administrador mediante un usuario y contraseña
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Número de requisito	RF-02
Nombre de requisito	Gestión de Clientes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requisito	Crear clientes nuevos Editar datos personales Dar de baja a un cliente
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Número de requisito	RF-03
Nombre de requisito	Gestión de Pagos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requisito	Registrar una suscripción con el total a pagar y lo que deje abonando el cliente además con una fecha límite. Realizar pagos de lo que le falte pagar por la suscripción al gimnasio. Realizar notificaciones de fin suscripción

Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Número de requisito	RF-04
Nombre de requisito	Gestión de Ejercicios y Alimentos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requisito	Crear, eliminar, modificar ejercicios y alimentos con sus valores nutricionales
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Número de requisito	RF-05
Nombre de requisito	Actualización de Base de Conocimiento
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requisito	<p>Descargar base de conocimiento</p> <p>Subir una nueva base de conocimiento</p>
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Número de requisito	RF-06
Nombre de requisito	Acceso a la App Móvil
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requisito	<p>Controlar el ingreso de usuarios mediante su correo y una contraseña</p> <p>El ingreso es únicamente a los usuarios registrados por el Administrador</p> <p>Los usuarios dados en baja, no les permitirá acceder a todo el sistema, pero si a sus datos personales</p>
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Número de requisito	RF-07
Nombre de requisito	Datos de los clientes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requisito	Los clientes accederán y editar todos sus datos mediante la aplicación móvil

Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Número de requisito	RF-08
Nombre de requisito	Generación de Rutinas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requisito	Los clientes mediante la aplicación móvil generarán rutinas diarias automáticamente evaluando los datos proporcionados a la aplicación. También tendrán la opción de crear su propia rutina
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Número de requisito	RF-09
Nombre de requisito	Generación de Dietas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requisito	Los clientes mediante la aplicación móvil generarán dietas diarias automáticamente evaluando los datos proporcionados a la aplicación. También tendrán la opción de crear su propia dieta
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Número de requisito	RF-10
Nombre de requisito	Gestión de Promociones
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Descripción del requisito	El administrador podrá crear, editar o eliminar promociones que visualizará el cliente en la fecha que se establezca
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

2.2.7 Requisitos No funcionales

Los requisitos no funcionales derivados del análisis de las historias de usuario se presentan en la Tabla 20:

TABLA 20. REQUISITOS NO FUNCIONALES

Número de requisito	RNF-01
Nombre de requisito	Interfaz y acceso al sistema
Descripción	El sistema para el administrador será accesible en cualquier navegador adaptando su interfaz para diferentes pantallas. La aplicación móvil será accesible en los dispositivos con Android 8.x o superior.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Número de requisito	RNF-02
Nombre de requisito	Disponibilidad del sistema
Tipo	<input checked="" type="checkbox"/> Requisito
Descripción	El sistema será accesible las 24 horas al día, 7 días a la semana.
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
Número de requisito	RNF-03
Nombre de requisito	Seguridad de la Información
Tipo	<input checked="" type="checkbox"/> Requisito
Descripción	El sistema será accesible mediante un usuario y contraseña en caso del administrador, y un email y contraseña en caso de la aplicación móvil, garantizando la seguridad de sus datos mediante una contraseña encriptada
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

2.3 Casos de uso

En los casos de uso se define las tareas que podrá cumplir la aplicación, así como también se pondrá los diferentes actores encargados de realizar las tareas.

2.3.1 Usuarios de la aplicación

Los usuarios que pueden acceder al sistema son el administrador a través de una interfaz web desde una computadora y el cliente a través de la aplicación móvil desde su celular. En la Figura 8 se presenta los usuarios actores en lenguaje UML.

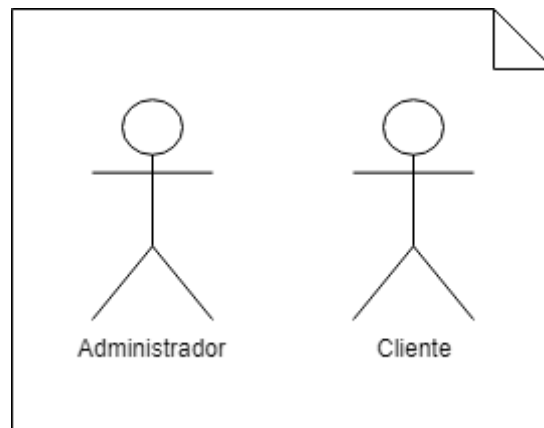


Figura 8. Usuarios de la aplicación

2.3.2 Caso de uso Administrador

Las acciones generales para el usuario “Administrador” se detallan e indican en la Figura 9 y se detallan en la Tabla 21:

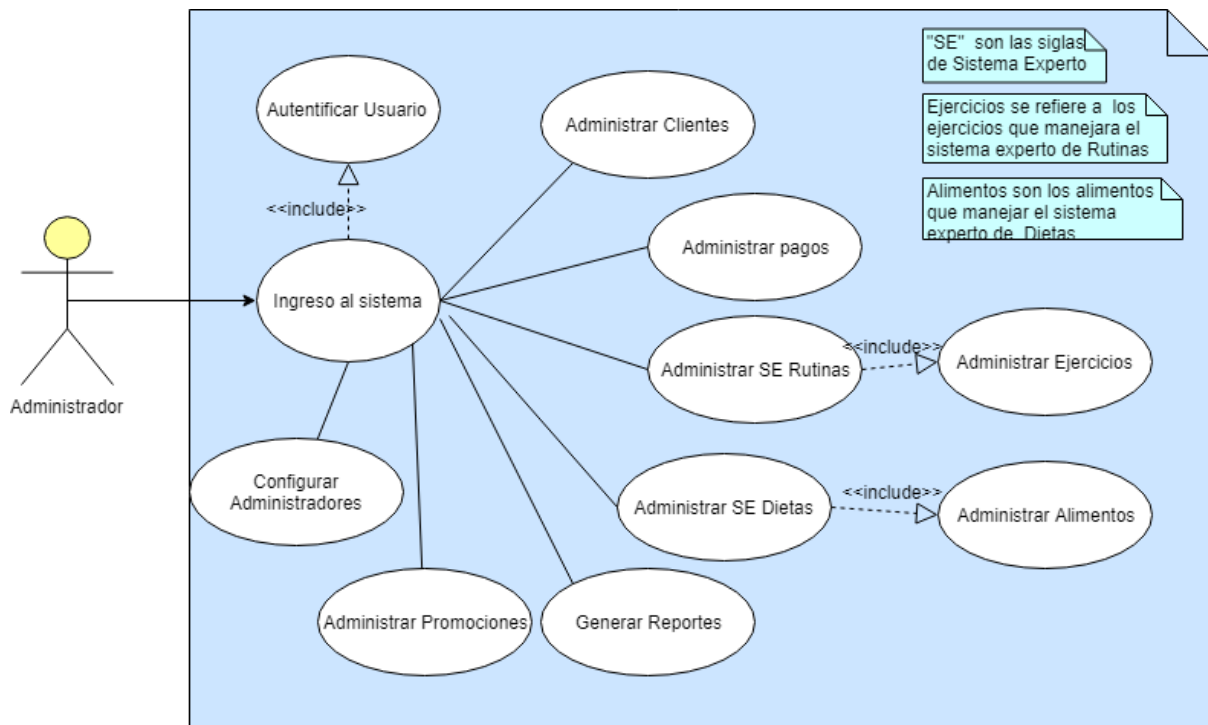


Figura 9. Caso de Uso Administrador

TABLA 21. DETALLE CASO DE USO ADMINISTRADOR

Código: CU-01	Nombre del Caso de Uso: Ingreso al portal Web
Objetivo	Controlar el ingreso de usuarios no permitidos

Descripción	Reconocer los datos del administrador al momento de ingresar al sistema	
Precondición	<p>Tener un ordenador con acceso a internet y a un navegador web.</p> <p>Tener registrado la información del administrador dentro del sistema</p>	
Secuencia	Paso	Acción
Normal	1	El usuario entrara con una dirección web del sistema
	2	Insertara su usuario y contraseña
	3	El sistema mostrara su interfaz de usuario administrador
	4	Fin del caso de uso
Postcondición	Las funcionalidades que puede acceder el administrador se encontraran en la interfaz del sistema	
Comentarios	Sin Comentarios	
Código: CU-02	Nombre del Caso de Uso: Autenticar Usuario	
Objetivo	Autenticar Administradores	
Descripción	Reconocer contraseña y usuario administrador; devolver los datos del administrador y permitir el acceso al sistema.	
Precondición	Ingresar al sistema	
Secuencia	Paso	Acción
Normal	1	EL usuario ingresa al sistema como administrador.
	2	Ingresar usuario y contraseña
	3	Se abre la interfaz del administrador
Postcondición	Se queda guardado los datos del administrador	
Comentarios	Sin Comentarios	
Código: CU-03	Nombre del Caso de Uso: Administrar Clientes	
Objetivo	Administrar los clientes del sistema	
Descripción	El sistema permitirá al administrador la gestión de clientes del gimnasio.	
Precondición	Debe estar iniciada la sesión de administrador	
Secuencia	Paso	Acción
Normal	1	El usuario accede al sistema como administrador
	2	El sistema despliega la interfaz del administrador
	3	En el menú seleccione clientes
	4	Administrar los clientes

	5	Fin del caso de uso
Postcondición	Se registraron los cambios en la información de los clientes dentro de la base de datos y en el sistema	
Comentarios	Sin Comentarios	
Código: CU-04	Nombre del Caso de Uso: Administrar Pagos	
Objetivo	Administrar los pagos de cada cliente registrado	
Descripción	El sistema permitirá al administrador acceder a la administración de los pagos.	
Precondición	Debe estar iniciada la sesión de administrador	
Secuencia	Paso	Acción
Normal	1	El usuario accede al sistema como administrador
	2	El sistema despliega la interfaz del Administrador
	3	En el menú seleccione Pagos
	4	Administrar los Pagos
	5	Fin del caso de uso
Postcondición	Los datos de los pagos se guardarán en el sistema	
Comentarios	Los pagos se los realizará por suscripción, la misma que deberá ser creada antes de realizar un pago.	
Código: CU-05	Nombre del Caso de Uso: Administrar SE de Rutinas	
Objetivo	Administrar los datos para la interacción del sistema experto encargado de dar rutinas a los clientes	
Descripción	El sistema permitirá al administrador crear ejercicios que se verán reflejados en la interfaz de rutinas del cliente además permite al usuario administrador actualizar la base de conocimiento.	
Precondición	Debe estar iniciada la sesión de administrador	
Secuencia	Paso	Acción
Normal	1	El usuario accede al sistema como administrador
	2	El sistema despliega la interfaz del Administrador
	3	En el menú seleccione Rutinas
	4	Administrar ejercicios y actualizar base de conocimiento
	5	Fin del caso de uso
Postcondición	Se registrarán los cambios de los ejercicios y la base de conocimientos en el sistema	

Comentarios	Sin Comentarios	
Código: CU-06	Nombre del Caso de Uso: Administrar Ejercicios	
Objetivo	Administrar los ejercicios registrados en el sistema	
Descripción	El sistema permitirá al administrador la gestión de los ejercicios registrados en el sistema	
Precondición	Debe estar iniciada la sesión de administrador	
Secuencia	Paso	Acción
Normal	1	El usuario accede al sistema como administrador
	2	El sistema despliega la interfaz del Administrador
	3	En el menú seleccione Rutinas
	4	Administrar ejercicios
	5	Fin del caso de uso
Postcondición	Se registrarán los cambios de los ejercicios en el sistema	
Comentarios	Sin Comentarios	
Código: CU-07	Nombre del Caso de Uso: Administrar SE Dietas	
Objetivo	Administrar los datos para la interacción del sistema experto encargado de dar dietas a los clientes	
Descripción	El sistema permitirá al administrador crear alimentos que se verán reflejados en la interfaz de rutinas del cliente además permite al usuario administrador actualizar la base de conocimiento.	
Precondición	Debe estar iniciada la sesión de administrador	
Secuencia	Paso	Acción
Normal	1	El usuario accede al sistema como administrador
	2	El sistema despliega la interfaz del Administrador
	3	En el menú seleccione Dietas
	4	Administrar alimentos y actualizar base de conocimiento
	5	Fin del caso de uso
Postcondición	Se registrarán los cambios de los alimentos y la base de conocimientos en el sistema	
Comentarios	Sin Comentarios	
Código: CU-08	Nombre del Caso de Uso: Administrar Alimentos	
Objetivo	Administrar los valores nutricionales de alimentos registrados en el sistema	

Descripción	El sistema permitirá al administrador la gestión de los alimentos con sus valores nutricionales que se encuentren registrados en el sistema	
Precondición	Debe estar iniciada la sesión de administrador	
Secuencia	Paso	Acción
Normal	1	El usuario accede al sistema como administrador
	2	El sistema despliega la interfaz del Administrador
	3	En el menú seleccione Dietas
	4	Administrar alimentos
	5	Fin del caso de uso
Postcondición	Se registrarán los cambios de los alimentos en el sistema	
Comentarios	Sin Comentarios	
Código: CU-09	Nombre del Caso de Uso: Generar Reportes	
Objetivo	Generar Reportes de los clientes del Gimnasio	
Descripción	El sistema permitirá al administrador generar reportes de los clientes registrados del gimnasio	
Precondición	Debe estar iniciada la sesión de administrador	
Secuencia	Paso	Acción
Normal	1	El usuario accede al sistema como administrador
	2	El sistema despliega la interfaz del Administrador
	3	En el menú lateral seleccione Reportes
	4	Seleccione el reporte a generar
	5	Fin del caso de uso
Postcondición	Se registrarán los cambios de los alimentos en el sistema	
Comentarios	Sin Comentarios	
Código: CU-10	Nombre del Caso de Uso: Configurar Administradores	
Objetivo	Configurar los datos de los administradores del sistema	
Descripción	El sistema permitirá al administrador autenticado, configurar sus datos como usuario y contraseña; además podrá crear nuevos administradores	
Precondición	Debe estar iniciada la sesión de administrador	
Secuencia	Paso	Acción
Normal	1	El usuario accede al sistema como administrador
	2	El sistema despliega la interfaz del Administrador
	3	En el menú lateral seleccione Configuración

	4	Configure datos de administradores
	5	Fin del caso de uso
Postcondición	Se registrarán los cambios de realizados de los administradores en el sistema	
Comentarios	Sin Comentarios	
Código: CU-10	Nombre del Caso de Uso: Administrar Promociones	
Objetivo	Administrar promociones para la visualización del cliente	
Descripción	El sistema permitirá al administrador crear una promoción para que sea visualizada por los que tengan instalada la aplicación móvil	
Precondición	Debe estar iniciada la sesión de administrador	
Secuencia	Paso	Acción
Normal	1	El usuario accede al sistema como administrador
	2	El sistema despliega la interfaz del Administrador
	3	En el menú lateral seleccione Promociones
	4	Administre promociones
	5	Fin del caso de uso
Postcondición	Se registrarán los datos de promociones en el sistema	
Comentarios	Sin Comentarios	

2.3.3 Casos de uso del Cliente

Las acciones generales para el usuario “Cliente” se detallan en la Figura 10 y se detallan en la Tabla 22:

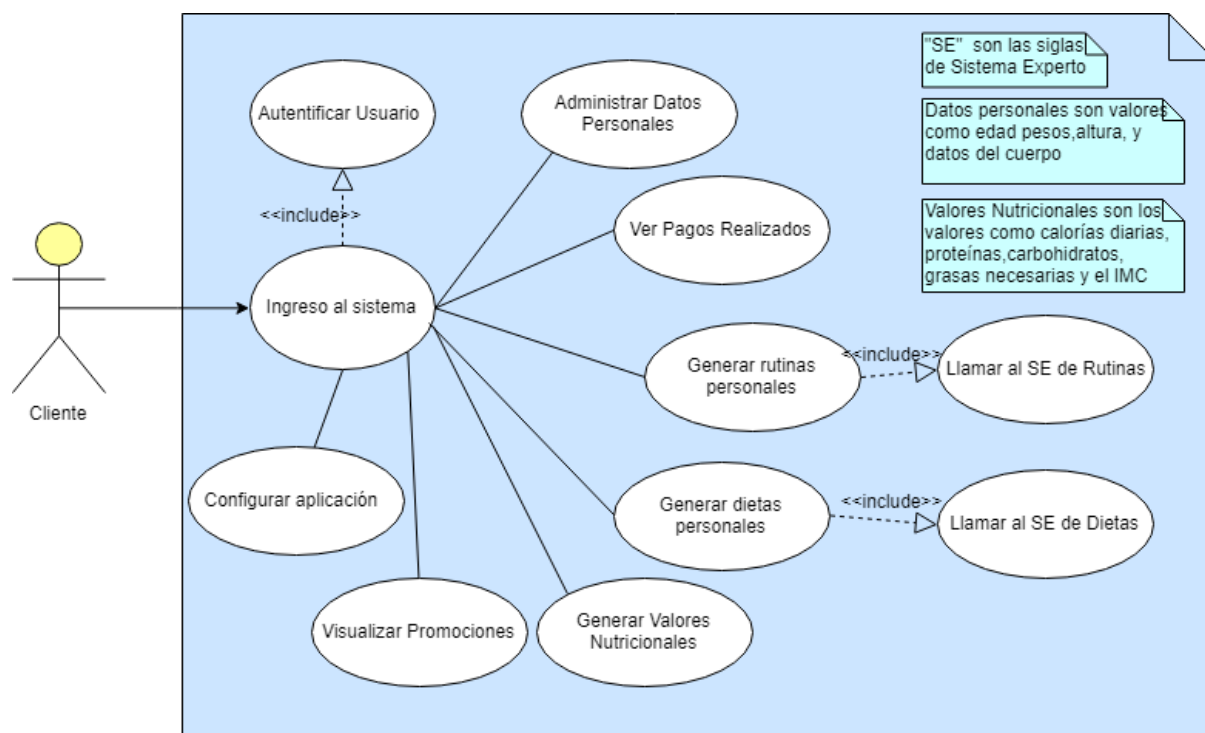


Figura 10. Caso de uso del Cliente

TABLA 22. DETALLE CASO DE USO DEL CLIENTE

Código: CU-11	Nombre del Caso de Uso: Ingreso al portal	
Objetivo	Controlar el ingreso de usuarios no permitidos	
Descripción	Reconocer los datos del cliente al momento de ingresar al sistema	
Precondición	Tener un teléfono con Android 6 o superior Tener instalada la aplicación del gimnasio en el teléfono	
Secuencia	Paso	Acción
Normal	1	El usuario ingresa a la aplicación a través de su teléfono
	2	Insertara su usuario y contraseña
	3	El sistema mostrara la interfaz del cliente junto a sus datos
	4	Guardar sesión en la memoria del móvil
	5	Fin del caso de uso
Postcondición	Las funcionalidades que puede acceder el cliente se encontraran en la interfaz del sistema	
Comentarios	Sin Comentarios	

Código: CU-12	Nombre del Caso de Uso: Administrar Datos Personales.	
Objetivo	Administrar datos personales por cliente	
Descripción	Los clientes podrán administrar sus datos personales como edad, peso, altura y así ver un progreso en sus datos dependiendo de lo que registren en el sistema	
Precondición	Debe estar abierta la aplicación y guardada la sesión en el móvil	
Secuencia	Paso	Acción
Normal	1	El usuario accede a la aplicación
	2	La aplicación despliega la interfaz del cliente
	3	Seleccione la segunda pestaña
	4	En el menú seleccione Datos y Medidas
	5	Administre sus datos personales
	6	Fin del caso de uso
Postcondición	Se guardará la información en el sistema y en la base de datos	
Comentarios	Sin Comentarios	
Código: CU-13	Nombre del Caso de Uso: Ver Pagos	
Objetivo	Visualizar los pagos de un cliente	
Descripción	El sistema permitirá al cliente visualizar su último pago para saber si necesitar acabar de pagar por la suscripción.	
Precondición	Debe estar abierta la aplicación y guardada la sesión en el móvil	
Secuencia	Paso	Acción
Normal	1	El usuario accede a la aplicación
	2	La aplicación despliega la interfaz del cliente
	3	Seleccione la segunda pestaña
	4	En el menú seleccione Pagos
	5	Visualice los pagos realizados
	6	Fin del caso de uso
Postcondición	Los datos de los pagos registrados por el administrador se presentarán en la aplicación	
Comentarios	Los pagos se los realizará por suscripción, la misma que deberá ser creada antes de ver un pago.	
Código: CU-14	Nombre del Caso de Uso: Generar Rutinas Personales	
Objetivo	El cliente deberá generar una rutina para realizar en el día	
Descripción	La aplicación dará opciones para la generación de rutinas para cada cliente	

Precondición	Debe estar abierta la aplicación y guardada la sesión en el móvil	
Secuencia	Paso	Acción
Normal	1	El usuario accede a la aplicación
	2	La aplicación despliega la interfaz del cliente
	3	Seleccione la segunda pestaña
	4	En el menú seleccione Rutinas
	5	Cree su rutina con las opciones de la aplicación
	6	Fin del caso de uso
Postcondición	Las rutinas del cliente se guardarán en el sistema para que tenga un registro	
Comentarios	Sin Comentarios	
Código: CU-15	Nombre del Caso de Uso: Llamar al SE de Rutinas	
Objetivo	Generar rutina automáticamente	
Descripción	La aplicación permitirá crear una rutina diaria automáticamente dependiendo lo que el usuario quiera realizar	
Precondición	Debe estar abierta la aplicación y guardada la sesión en el móvil además debe estar registrados datos personales del usuario	
Secuencia	Paso	Acción
Normal	1	El usuario accede a la aplicación
	2	La aplicación despliega la interfaz del cliente
	3	Seleccione la segunda pestaña
	4	En el menú seleccione Rutinas
	5	Seleccione Sistema Experto
	6	Visualice y guarde la rutina si está de acuerdo
	7	Fin del caso de uso
Postcondición	Se registrarán los cambios de las rutinas del cliente en el sistema	
Comentarios	Sin Comentarios	
Código: CU-16	Nombre del Caso de Uso: Generar Dietas Personales	
Objetivo	El cliente deberá generar una dieta a seguir en el día	
Descripción	La aplicación dará opciones para la generación de dietas para cada cliente dependiendo sus valores nutricionales	
Precondición	Debe estar abierta la aplicación y guardada la sesión en el móvil	
Secuencia	Paso	Acción

Normal	1	El usuario accede a la aplicación
	2	La aplicación despliega la interfaz del cliente
	3	Seleccione la segunda pestaña
	4	En el menú seleccione Dietas
	5	Cree su dieta con las opciones de la aplicación
	6	Fin del caso de uso
Postcondición	Las rutinas del cliente se guardarán en el sistema para que tenga un registro	
Comentarios	Sin Comentarios	
Código: CU-17	Nombre del Caso de Uso: Llamar al SE de Dietas	
Objetivo	Generar dieta automáticamente	
Descripción	La aplicación permitirá crear una dieta diaria automáticamente dependiendo los valores nutricionales que el usuario haya ingresado	
Precondición	Debe estar abierta la aplicación y guardada la sesión en el móvil además tiene que estar registrados datos personales del cliente	
Secuencia	Paso	Acción
Normal	1	El usuario accede a la aplicación
	2	La aplicación despliega la interfaz del cliente
	3	Seleccione la segunda pestaña
	4	En el menú seleccione Dietas
	5	Seleccione Sistema Experto
	6	Visualice y guarde la dieta si está de acuerdo
	7	Fin del caso de uso
Postcondición	Se registrarán los valores de la dieta por cliente en el sistema	
Comentarios	Sin Comentarios	
Código: CU-18	Nombre del Caso de Uso: Generar Valores Nutricionales	
Objetivo	Generar valores nutricionales del cliente	
Descripción	La aplicación permitirá al cliente visualizar sus valores nutricionales dependiendo sus datos personales	
Precondición	Debe estar abierta la aplicación y guardada la sesión en el móvil además tiene que estar registrados datos personales del cliente	
Secuencia	Paso	Acción
Normal	1	El usuario accede a la aplicación

	2	La aplicación despliega la interfaz del cliente
	3	Seleccione la segunda pestaña
	4	En el menú seleccione Datos calculados
	5	Cree su dieta con las opciones de la aplicación
	6	Fin del caso de uso
Postcondición	Los datos se generarán automáticamente una vez que se registra los datos personales	
Comentarios	Estos valores nutricionales se guardan en la base de datos y cambian por cada ingreso nuevo de datos	
Código: CU-19	Nombre del Caso de Uso: Configurar Aplicación	
Objetivo	Permitir al usuario cambiar parámetros	
Descripción	La aplicación permitirá al usuario cambiar contraseña y cerrar sesión, además podrá editar la intensidad de esfuerzo para realizar sus valores nutricionales	
Precondición	Debe estar abierta la aplicación y guardada la sesión en el móvil además tiene que estar registrados datos personales del cliente	
Secuencia	Paso	Acción
Normal	1	El usuario accede a la aplicación
	2	La aplicación despliega la interfaz del cliente
	3	Seleccione la tercera pestaña
	4	Realice configuraciones
	5	Fin del caso de uso
Postcondición	Se registrarán los cambios de realizados por el cliente en el sistema	
Comentarios	Sin Comentarios	
Código: CU-20	Nombre del Caso de Uso: Visualizar Promociones	
Objetivo	Permitir al usuario ver promociones creadas por el administrador	
Descripción	La aplicación permitirá a los clientes del gimnasio visualizar las promociones disponibles que haya creado el administrador	
Precondición	Debe estar abierta la aplicación y guardada la sesión en el móvil además tiene que estar registrada información de promociones	
Secuencia	Paso	Acción
Normal	1	El usuario accede a la aplicación
	2	La aplicación despliega la interfaz del cliente
	3	Seleccione la segunda pestaña
	4	En el menú selección promociones

	6	Visualice las promociones
	6	Fin del caso de uso
Postcondición	En la aplicación se cargará una interfaz para visualizar las promociones	
Comentarios	Sin Comentarios	

2.4 Diseño

2.4.1 Diseño conceptual

El Modelo Conceptual o también llamado diagrama de Modelo de Dominio permite demostrar los conceptos más relevantes y sus asociaciones para el sistema. En la Figura 11 se presenta el diagrama conceptual, que se desarrolló a partir de un análisis de las historias de usuario, el cual se tomara en cuenta para la realización de la base de datos en PostgreSQL.

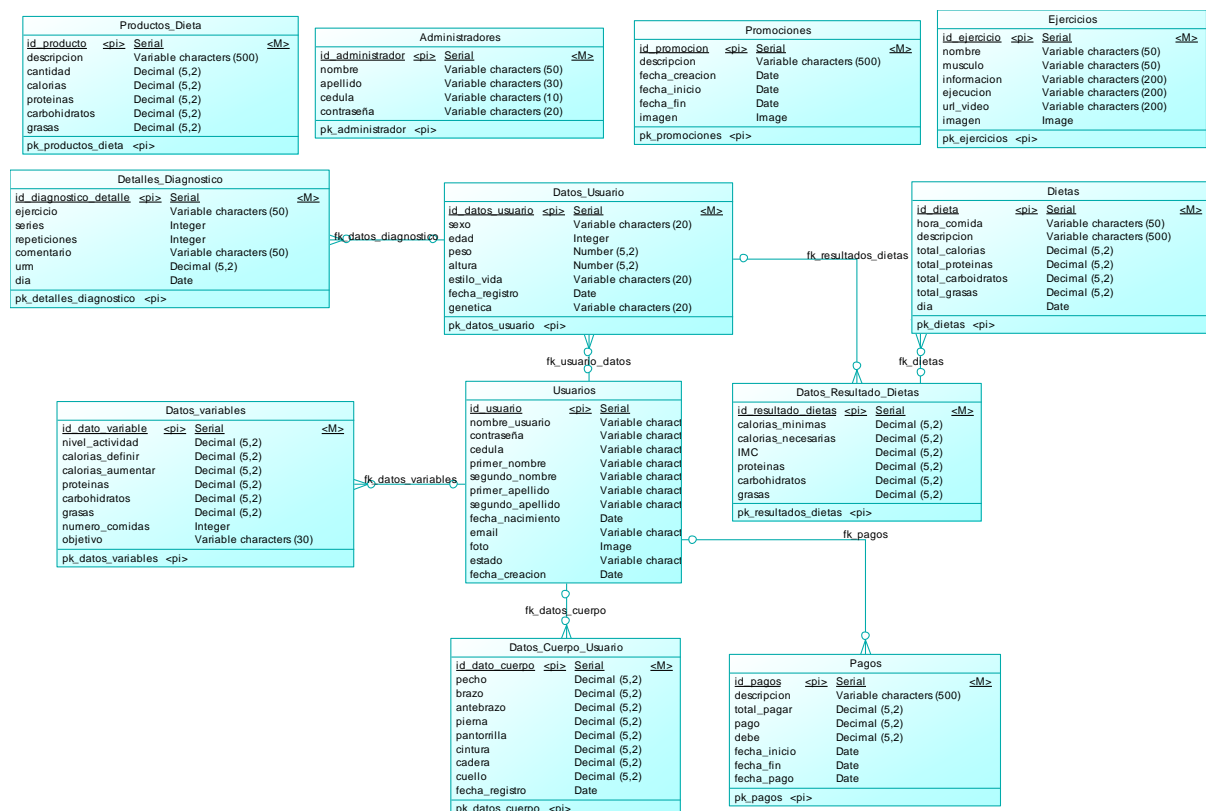


Figura 11. Diseño conceptual del sistema

2.4.2 Diagrama de clases

El diagrama de clases permite demostrar en forma general la relación e interacción que tiene las diferentes clases del sistema. En la Figura 12 se presenta el diagrama de clases que se desarrolló a partir de un análisis de las historias de usuario, el cual se tomará en cuenta para la realización del modelo de clases en Spring Framework.

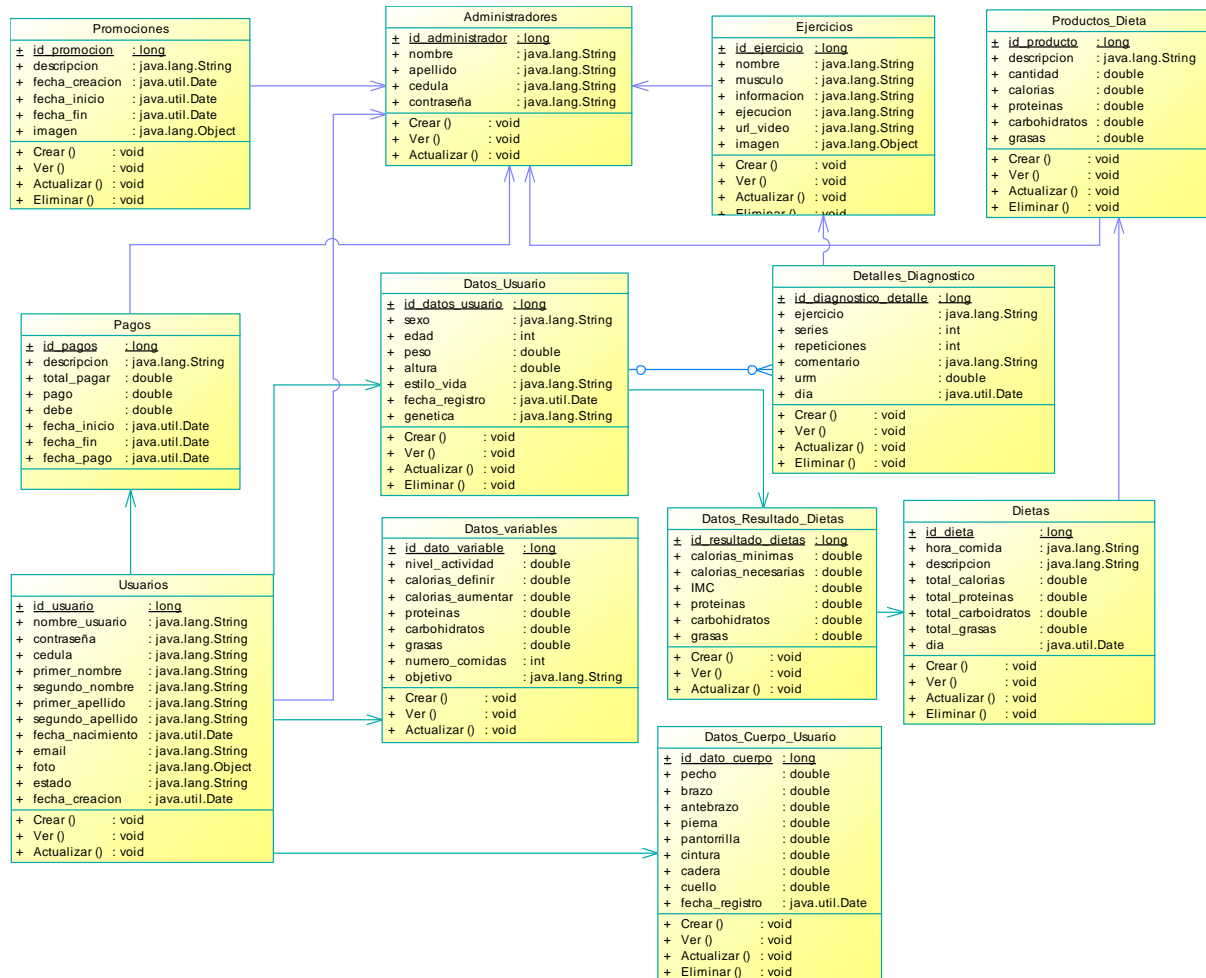


Figura 12. Diseño de clases del sistema

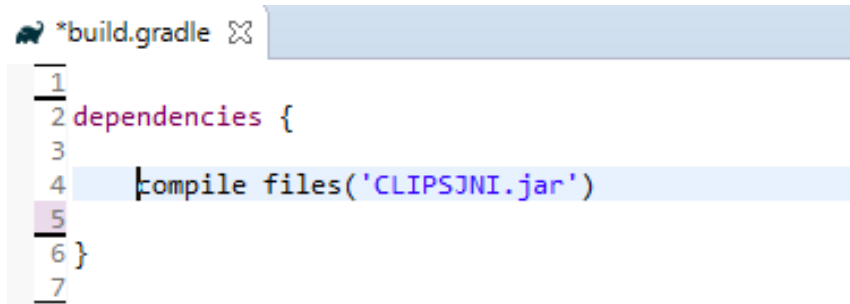
2.4.3 Integración de CLIPS con Spring

Para el desarrollo y producción del sistema experto se ha elegido la herramienta Clips y para la publicación de servicios y ejecución de tareas se lo ha realizado con el framework Spring del lenguaje de programación java.

La herramienta Clips originalmente esta desarrollado en lenguaje C por lo que la integración con ese lenguaje es mucho más sencilla, pero también ofrece un JNI (Java Native Interface) que es un framework que permite a CLIPS interactuar con java.

Para instalar el JNI es sencillo ya que podemos importarlo como librería con JAVA, pero en el caso de Spring será necesario implementarlo en el Gradle del proyecto Spring para que

llame a la librería necesaria en el momento de compilación. A continuación, en la Figura 13 se detalla el código necesario para integrar Spring y el JNI de CLIPS:

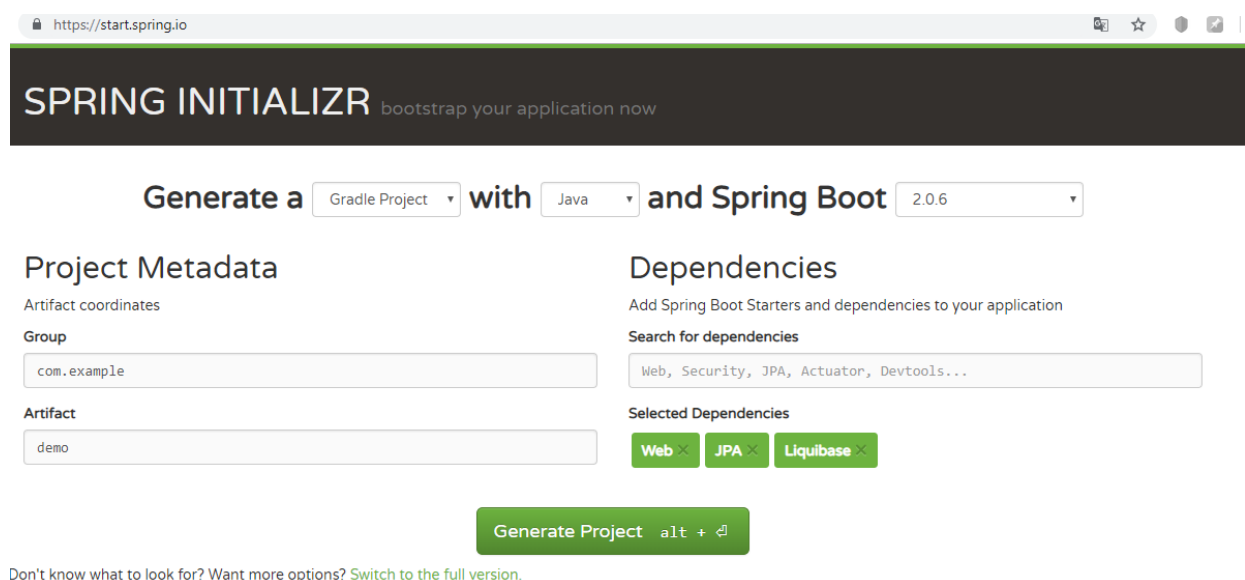


```
*build.gradle
1
2 dependencies {
3
4     compile files('CLIPSJNI.jar')
5
6 }
7
```

Figura 13. Referencia del código para integrar CLIPS

2.4.4 Integración de Spring y Angular JS

Debido que el sistema está basado en una arquitectura cliente servidor, se utilizó Spring framework como el servidor de la aplicación debido que ya ahorra tiempo en realizar las configuraciones para exponer Servicios Rest en java. El patrón de diseño a seguir en Spring es MVC y a través de la página web <https://start.spring.io/> presentado en la Figura 14, se puede crear un modelo de las clases y carpetas necesarias para empezar un proyecto con Spring



https://start.spring.io

SPRING INITIALIZR bootstrap your application now

Generate a Gradle Project with Java and Spring Boot 2.0.6

Project Metadata

Artifact coordinates

Group

Artifact

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies

Selected Dependencies

Web JPA Liquibase

Generate Project alt + ⌘

Don't know what to look for? Want more options? [Switch to the full version.](#)

Figura 14. Creación del modelo de Spring

La estructura de un proyecto en Spring framework con sus directorios necesarios se detalla en la Figura 15:

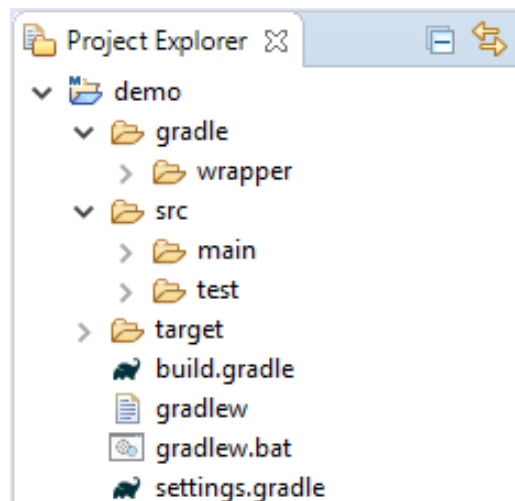


Figura 15. Estructura de un proyecto con Spring

Por otro lado, en el apartado del cliente de la arquitectura que se está siguiendo se ejecuta el framework Angular JS para el usuario Administrador, el cual está constituido por varias herramientas de desarrollo web como HTML5, CSS3, JavaScript y TypeScript las que en conjunto permitirá crear una interfaz para el cliente final interactuando con los servicios web de Spring. Para la ejecución e instalación de angular es necesario instalar primero la herramienta NodeJS, y una vez instalado angular tendrá una estructura como la Figura 16:

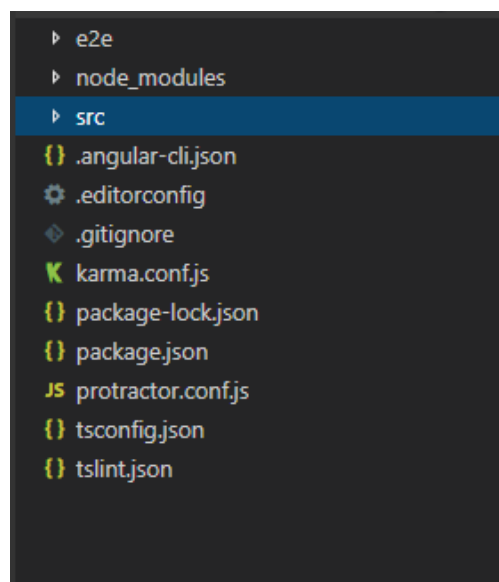


Figura 16. Estructura del proyecto en Angular

Como Spring framework se expondrá los servicios web para intercambiar los datos con cualquier cliente que desee acceder mediante protocolos web, angular provee de un módulo llamado HttpClient el cual provee los métodos necesarios para llamar a estos servicios y monitorizar los datos que envíe o reciba el servidor, desde este framework.

2.4.5 Integración de Spring y Ionic JS

Como Ionic framework utiliza el mismo Angular para crear una aplicación móvil híbrida, se puede ejecutar el mismo proceso de la integración de angular con Spring para la llamada de los servicios web y comunicarse con el servidor para visualizar o modificar los datos del sistema. El módulo HttpClient de igual manera viene instalado con Ionic y se puede llamar a los servicios web de Spring Framework.

2.5 Iteraciones

En la metodología XP para que el cliente este conforme con lo acordado en las historias de usuario se le muestra avances del sistema cada cierto tiempo lo cual se le llama iteraciones en la metodología lo que permite al cliente seguir de cerca el proyecto, corregir fallas o modificar partes del sistema para asegurar que el sistema se encuentre cumpliendo con las expectativas del cliente. En cada historia de usuario se especificó a que iteración pertenece cada una, así como se demuestra en la Tabla 23 con un resumen de las historias de usuario y las iteraciones a realizar:

TABLA 23. DESCRIPCIÓN DE ITERACIONES

Iteración	Historia de Usuario		Riesgo en el desarrollo
Primera Iteración	01. Acceso al Portal Web		Bajo
	02. Registrar Clientes		Bajo
	03. Administrar pago de clientes		Medio
	04. Notificar Suscripción	Fin	Medio
	05. Promocionar ofertas		Medio
	06. Reporte Clientes		Bajo

Segunda Iteración	07. Aplicación móvil	Medio
	08. Datos de Clientes	Medio
	09. Generar valores nutricionales	Alto
	10. Pago pendiente del cliente	Medio
	11. Registrar Rutinas del Cliente	Medio
	12. Registrar Dietas del Cliente	Alto
Tercera Iteración	13. Registrar Ejercicios y Alimentos	Bajo
	14. Actualizar Sistema Experto	Alto
	15. Determinación de rutinas y dietas del Sistema Experto	Alto

2.6 Primera Iteración

En esta iteración se tomó en cuenta el desarrollo de la interfaz Web en Angular para el usuario administrador y también se crearon las funciones principales del mismo, excluyendo la parte del sistema experto.

2.6.1 Historia de Usuario - Acceso al portal web

En la primera historia, se estableció que el cliente desea tener acceso a un portal web mediante un usuario y contraseña desde cualquier navegador para la administración del sistema. Por lo cual se realizó las siguientes acciones para el desarrollo de este:

- **Creación de Interfaz de Autenticación**

Se desarrolló la interfaz de acceso al sistema con campos usuario y contraseña requeridos para el ingreso al sistema del administrador como se indica en la Figura 17.

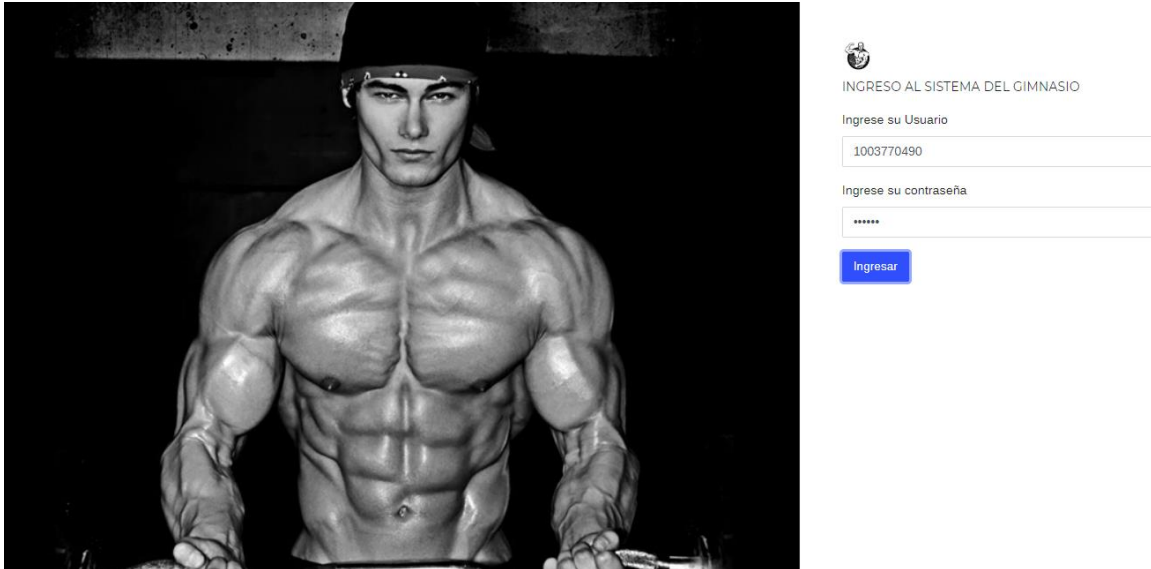


Figura 17. Interfaz de autenticación

- **Encriptación de la contraseña**

Cumpliendo con los requisitos de seguridad, se encripta las contraseñas guardadas del administrador en la base de datos

- **Bloqueo al acceso a las demás interfaces del administrador**

Se bloquea el acceso a las interfaces del sistema sin el previo ingreso o registro del usuario Administrador

- **Guardar Sesión y recibir notificaciones**

Al acceder al sistema se guardará la sesión del usuario, para evitar estar validando a cada momento al administrador. También recibirá las notificaciones del sistema al momento de acceder.

- **Interfaz de acceso al Usuario Administrador**

Se desarrollo una interfaz como se indica en la Figura 18 para acceder a las tareas principales de forma automática, con un menú lateral para ver todas las opciones que tiene el sistema y cerrar la sesión de manera sencilla.

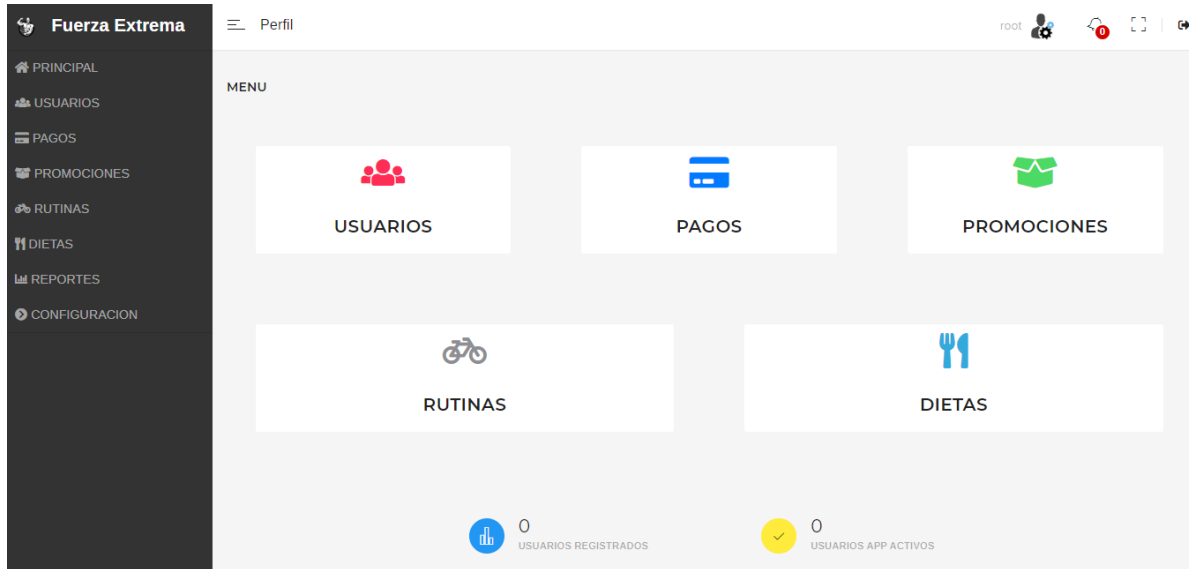


Figura 18. Interfaz al acceder el usuario

2.6.2 Historia de Usuario - Registrar clientes

Las acciones llevadas a cabo en la historia de usuario dos, para el registro de clientes se detallan a continuación:

- **Interfaz para administración de clientes**

Se creó una interfaz para el administrador como se indica en la Figura 19 para visualizar los usuarios que estén registrados en el sistema y ver algunos datos de estos.

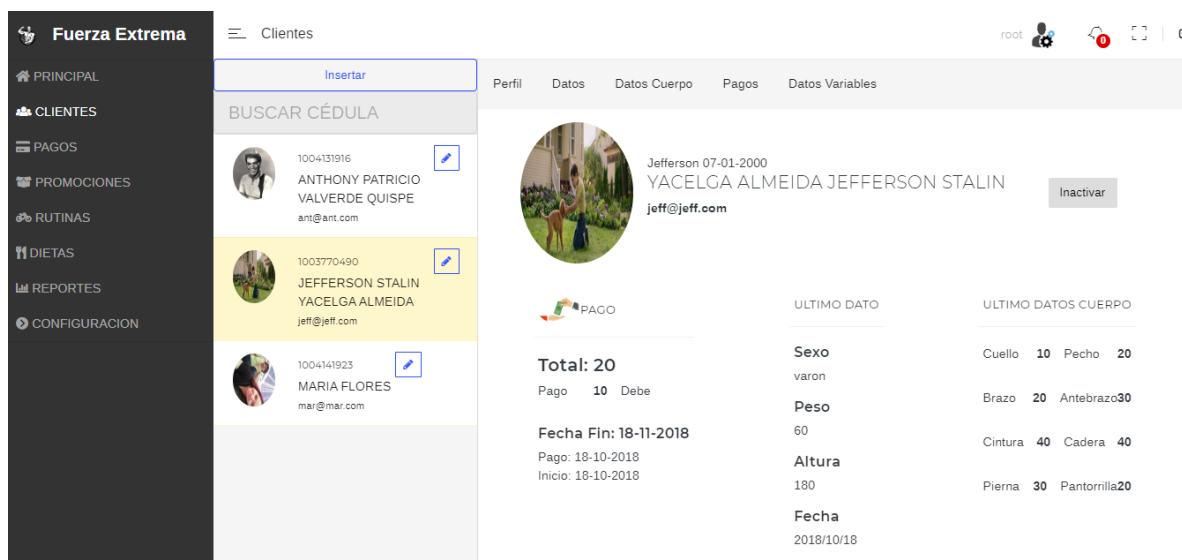


Figura 19. Interfaz de administración de clientes

- **Validación de datos**

Se desarrolló los métodos para validación y tratamiento de la información que envía el servidor y hacia la base de datos de los Clientes del gimnasio. Se valida que la cedula ingresada sea una cedula válida en Ecuador y validación de campos obligatorios que se deberán llenar para la creación de un usuario.

- **Dar de baja a un cliente**

Se desarrolló un método que permite dar de baja al cliente, en donde cambiará el estado del cliente y no le permitirá el uso completo de la aplicación móvil.

- **Servicios Web de Clientes**

Se crearon servicios para la visualización, acceso a datos de clientes, modificación y eliminación de clientes en la base de datos. Los cambios se verán reflejados en la aplicación de cada cliente.

2.6.3 Historia de usuario - Administrar pago de clientes

Para la historia sobre la administración de pagos de cada cliente, se realizó las siguientes acciones:

- **Creación de Interfaz para administrar pagos**

Se desarrolló una interfaz para el administrador como se indica en la Figura 20 en la cual el podrá ingresar la cedula de un cliente ya registrado en el sistema y ver todos sus pagos en una tabla.

Id	Descripcion	Fecha Registro	Fecha Inicio	Fecha Fin	Total Pagar	Pago
12	Plan Mensual	2018/10/18	2018/10/18	2018/11/18	20	2
11	Plan Mensual	2018/10/18	2018/10/18	2018/11/18	20	3

Figura 20. Interfaz de administración de pagos

- **Validación de Datos**

Cada pago que se registre en el sistema se lo puede hacer únicamente si el cliente tiene algún valor pendiente por su suscripción o plan en el gimnasio, en caso de no tener se da la opción de crear un nuevo plan de pago. Al administrador se le ofrece opciones ya preconfiguradas para crear el plan de cada cliente. Además, por cada pago se valida que no supere la cantidad adeudada desactivando el botón que guarda el pago.

- **Servicios Web de Clientes**

Se crearon servicios web en la parte del servidor para la visualización, acceso a datos de clientes y realizar pagos por cada cliente. Los cambios se verán reflejados en la aplicación de cada cliente en el apartado de pagos.

2.6.4 Historia de usuario - Notificar fin suscripción

Para la historia de usuario de la notificación por fin de suscripción se realizó las siguientes acciones:

- **Iniciar sesión y recibir notificaciones**

Se desarrolló un método en el servidor, que cuando se inicie la sesión de administrador se devuelva como respuesta si algún usuario está próximo a finalizar su suscripción, como también se notifica los usuarios que ya finalizaron su suscripción, pero no se encuentran dados de baja por el administrador

- **Interfaz para notificaciones**

Se desarrolló una pequeña interfaz, en el formulario principal del administrador en la parte superior derecha que permitirá visualizar las notificaciones cuando se abra la pestaña. En la Figura 21 se puede ver la interfaz que tendrán las notificaciones del sistema.

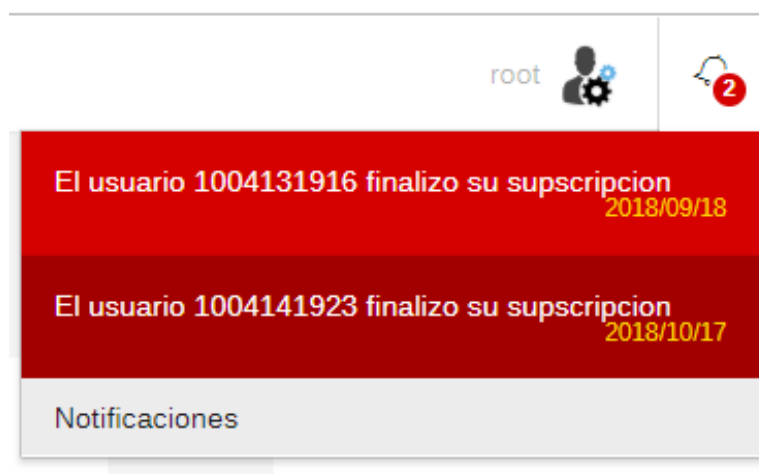


Figura 21. Interfaz para notificaciones

2.6.5 Historia de usuario - Promocionar ofertas

Para la historia de usuario sobre la promoción de ofertas del administrador, se realizó las siguientes acciones:

- **Creación de interfaz de ofertas**

Se desarrolló en la interfaz del administrador un apartado para configurar las ofertas que el desee implementar y sean visibles para los clientes en la aplicación móvil. En la Figura 22 se puede visualizar la interfaz de ofertas.

ID	Descripción	Fecha Creación	Fecha Inicio	Fecha Fin	Imagen
1	e35	11-09-2018	13-09-2018	13-09-2018	Ver

Figura 22. Interfaz para administrar ofertas

- **Creación de métodos para manejo de promociones**

Se desarrolló los métodos para añadir, editar y eliminar para que se pueda manejar los datos de las promociones dándoles una fecha de duración de la promoción para ser reflejada en la aplicación móvil.

- **Servicios Web de Promociones**

Se crearon servicios web en la parte del servidor para la visualización, acceso a datos de promociones y su manejo de datos con la base de datos. Los cambios se verán reflejados en la aplicación de cada cliente en el apartado de promociones.

2.6.6 Historia de usuario – Reportes

La historia de usuario reportes en la que se estableció la realización de reportes de la aplicación junto al usuario se realizó mediante las siguientes acciones

- **Creación de interfaz para realizar Reportes**

Se desarrolló una interfaz de reportes presentada en la Figura 23 que puede ser accesible mediante el menú lateral del sistema, en la cual se muestra las dos opciones de reportes que puede realizar el usuario.

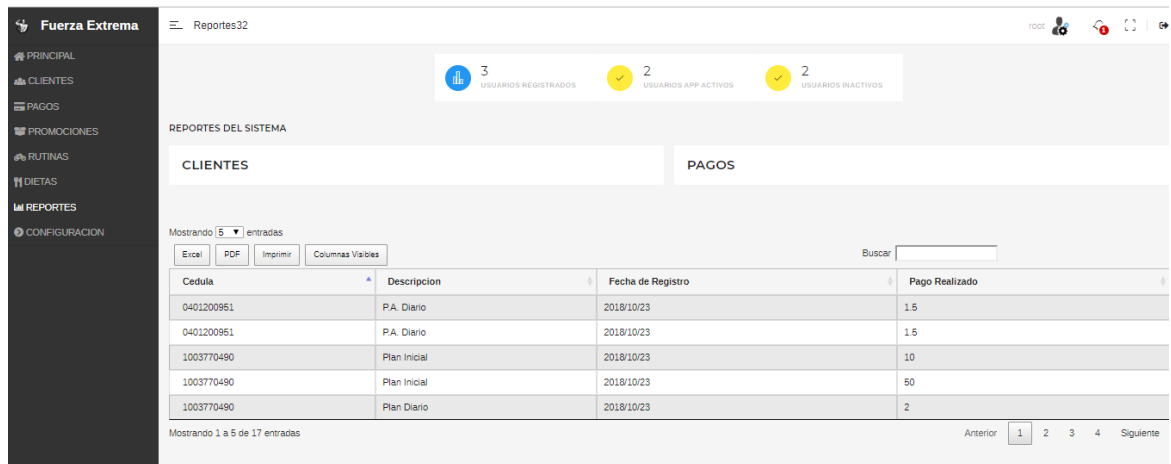


Figura 23. Interfaz de reportes

- **Tabla para visualizar fechas de registro de los usuarios**

El reporte acordado con el usuario, para que se pueda visualizar mediante una tabla los datos importantes del usuario, así mismo con su respectiva fecha de registro, se la obtuvo de la fecha que se guarda al momento de ingresar un nuevo usuario por parte del administrador.

- **Tabla para visualizar los usuario activos e inactivos**

Se desarrolló una tabla para visualizar los datos de los usuarios que siguen asistiendo al gimnasio y controlando el cumplimiento de pagos de los servicios que ofrece el gimnasio. También una tabla para identificar los usuarios que dejaron de asistir o realizar sus pagos correctamente.

- **Tabla para visualizar los pagos realizados por los usuarios**

Se añadió también una tabla de pagos realizados por cliente, donde se visualizará su respectiva cedula con la fecha de registro del pago realizado.

- **Exportación mediante PDF y Excel**

Las tablas realizadas en la interfaz de reportes fueron desarrolladas mediante la integración de jQuery para integrar funciones como paginación, búsqueda, selección de columnas para su visualización además la posibilidad de exportar las tablas a PDF y Excel, o incluso imprimir las tablas desde el navegador.

2.7 Segunda Iteración

Para la realización de esta iteración se tomó en cuenta los datos del cliente y la interfaz gráfica del mismo, el cual según el usuario se lo realizará como una aplicación móvil en donde se manejará los datos cada cliente.

2.7.1 Historia de usuario - Aplicación móvil

La historia de usuario 7 se centra en la realización de la aplicación móvil para los clientes del gimnasio por lo que se realizó las siguientes acciones:

- **Creación de la interfaz móvil**

Se desarrolló una interfaz junto a Ionic framework que consta de 3 pestañas principales presentadas en la Figura 24 donde se indica: los datos del cliente, las diferentes opciones de la aplicación y las configuraciones que se puede realizar en la misma.

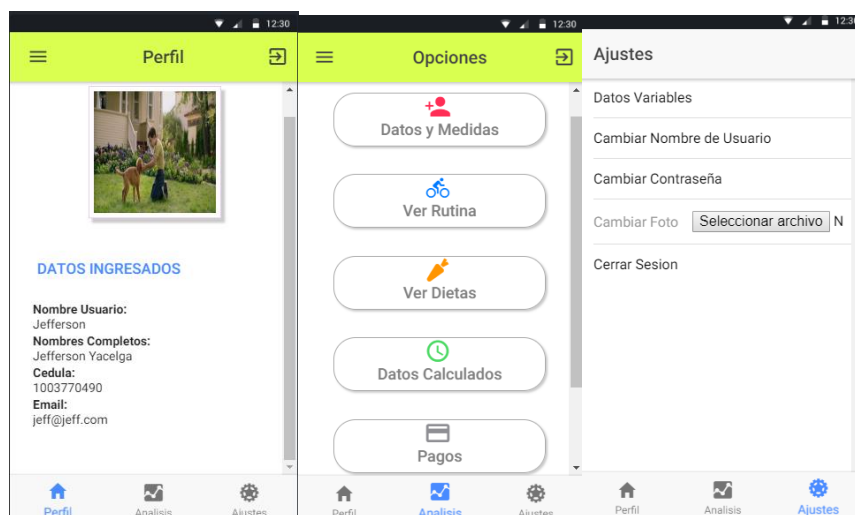


Figura 24. Interfaces del aplicativo móvil

- **Validación del cliente**

Para validación de los datos del cliente existe una interfaz para que el cliente ingrese su correo y contraseña presentada en la Figura 25. Se escogió como dato para

validación el correo ya que la mayoría de las aplicaciones solicitan el mismo para el acceso de sus datos y de esta forma es más fácil para el usuario recordar su usuario.



Figura 25. Interfaz de validación de clientes

- **Cambiar contraseña al primer ingreso**

Cumpliendo con el requisito de seguridad, la contraseña de cada cliente se crea por defecto por el administrador y por lo tanto el cliente al momento de ingresar deberá realizar un cambio de contraseña y podrá acceder a sus datos.

- **Guardar sesión**

La aplicación del cliente permitirá guardar los datos de la sesión al momento de iniciar sesión para evitar que ingrese los datos cada que acceda a la aplicación.

2.7.2 Historia de usuario - Datos de clientes

Para el acceso a los datos por cliente de la aplicación móvil según la historia de usuario 8, se realizó las siguientes acciones:

- **Creación de interfaz de acceso y edición de datos**

Se realizó una interfaz para que el cliente pueda visualizar todos sus datos personales registrados en la aplicación, para pasar al modo de edición en la parte superior se puso un botón que al interactuar con el mismo cambiara a modo edición y podrá crear nuevos datos. En la Figura 26 se indica la interfaz de los datos de clientes y el modo edición que dispone.

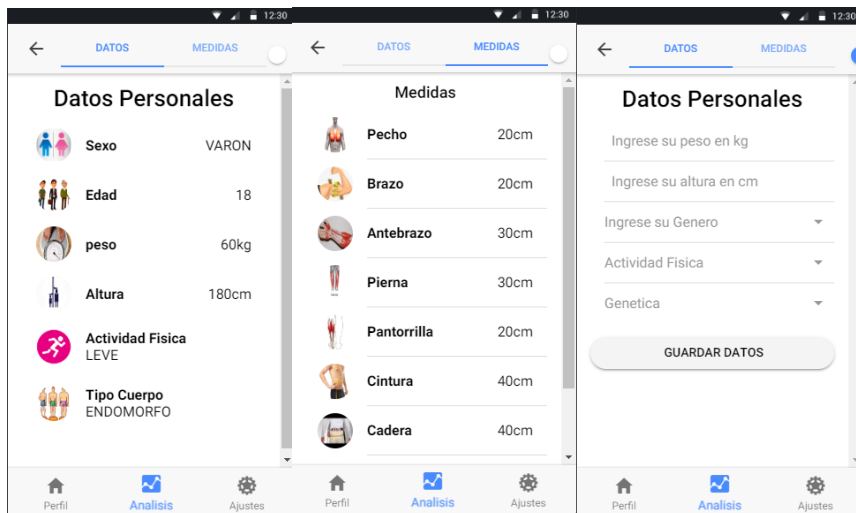


Figura 26. Interfaces de datos del cliente en el aplicativo móvil

- **Creación de servicios Web**

Se desarrolló servicios web en la parte del servidor para la visualización, acceso y creación a datos personales de cada cliente. Los cambios se verán reflejados en la aplicación de cada cliente en el apartado de promociones.

2.7.3 Historia de usuario - Generar valores nutricionales

Para la historia de usuario dirigida hacia la generación de valores nutricionales para cada cliente se realizaron las siguientes acciones:

- **Creación de interfaz para valores nutricionales**

Se desarrolló una interfaz en la aplicación móvil presentada en la Figura 27, para que el cliente pueda visualizar los valores nutricionales como IMC, calorías necesarias, calorías mínimas, proteínas, carbohidratos y grasas recomendadas.

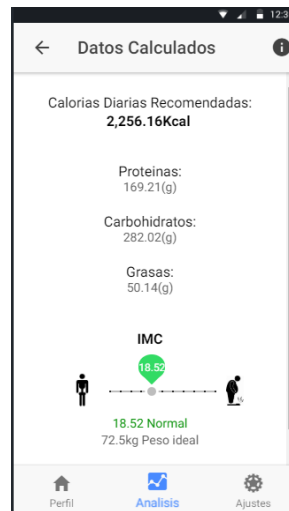


Figura 27. Interfaz de valores nutricionales

- **Creación de funciones para cálculos nutricionales**

La función principal para sacar las calorías necesarias o tasa basal está basada en la fórmula de Mifflin St. Jeor que se usa en el campo de nutrición ya que es mucho más certera en sus datos (Treviño, 2012). A continuación, se presenta las fórmulas empleadas para determinar todos los valores nutricionales del cliente:

$$\text{Calorías Min. Hombres} = 9.99 \times \text{Peso(kg)} + 6.25 \times \text{Altura} - 4.92 \times \text{Edad} - 161;$$

$$\text{Calorías Min. Mujeres} = 9.99 \times \text{Peso(kg)} + 6.25 \times \text{Altura(cm)} - 4.92 \times \text{Edad} + 5;$$

$$\text{Calorías necesarias} = (\text{Calorías mínimas} \times \text{nivel de actividad});$$

$$\text{Objetivo Mantenerse} = \text{calorías necesarias};$$

$$\text{Objetivo Aumentar} = (10 \text{ a } 30 \%) \text{ de calorías necesarias};$$

$$\text{Objetivo Definir} = - (10 \text{ a } 30 \%) \text{ de calorías necesarias};$$

$$\text{IMC} = \text{Peso (kg)} / (\text{Altura(m)})^2$$

$$\text{Peso ideal hombres} = \text{estatura (cm)} - 100 - [(\text{estatura(cm)} - 150) / 4];$$

$$\text{Peso ideal mujeres} = \text{estatura (cm)} - 100 - [(\text{estatura(cm)} - 150) / 2.5];$$

2.7.4 Historia de usuario - Pago pendiente del cliente

Conforme a la historia de usuario que se recopiló, para el pago pendiente del cliente se realizó las siguientes acciones:

- **Creación de interfaz de último pago**

Se desarrolló una interfaz en la aplicación móvil de los clientes para pagos presentada en la Figura 28, para que cada cliente pueda visualizar su último pago.

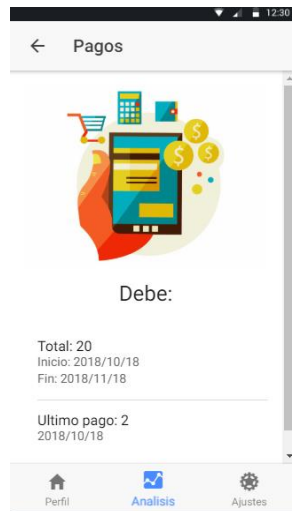


Figura 28. Interfaz de pagos en el aplicativo móvil

- **Creación de método para último pago**

Se desarrolló un método en el servidor para crear un servicio web que devuelva únicamente el último pago realizado por cada cliente

2.7.5 Historia de usuario - Registrar Rutinas del Cliente

Para la historia de usuario del registro de rutinas para el cliente, se realizó las siguientes acciones:

- **Crear interfaz para creación de rutinas**

Se desarrollo una interfaz en la aplicación móvil del cliente para que cada cliente pueda crear una rutina propia, proporcionándole una lista de ejercicios que el administrador debe registrar previamente.

- **Creación de métodos de acceso a datos del servidor**

En el servidor se desarrolló métodos para acceso a los datos de ejercicios y rutinas para cada cliente, además de método para creación de rutina por cada cliente y registrar en el sistema.

2.7.6 Historia de usuario - Registrar dietas del cliente

Para la historia de usuario del registro de dietas para el cliente, se realizó las siguientes acciones:

- **Crear interfaz para creación de dietas**

Se desarrolló una interfaz en la aplicación móvil del cliente presentada en la Figura 29 para que cada cliente pueda crear una dieta propia, proporcionándole una lista de alimentos con sus valores nutricionales que el administrador debe registrar previamente.

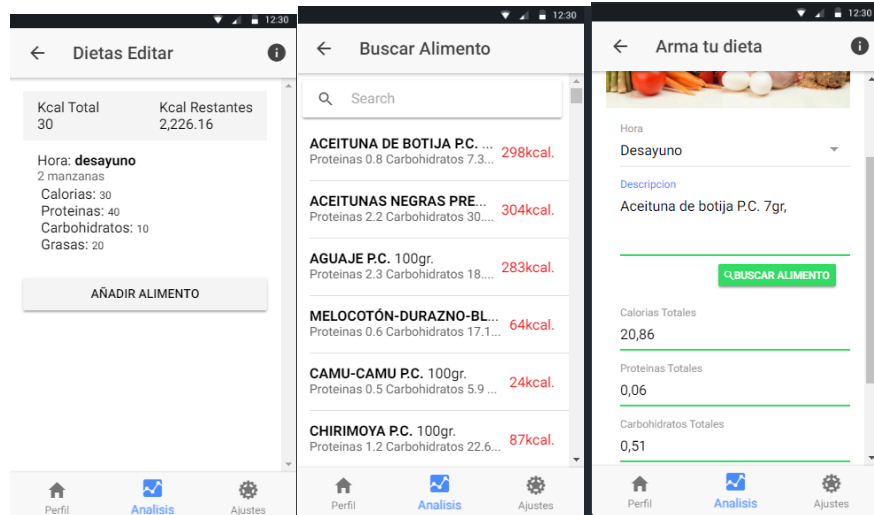


Figura 29. Interfaces para creación de dietas en el aplicativo móvil

- **Creación de métodos de acceso a datos del servidor**

En el servidor se desarrolló métodos para acceso a los datos de ejercicios y rutinas para cada cliente, además de método para creación de rutina por cada cliente y registrar en el sistema.

- **Calcular Calorías de los Alimentos**

En la aplicación móvil del cliente se desarrolló un método que suma los valores nutricionales de los alimentos para que el cliente tenga una idea de lo que debe consumir por cada hora del día. Se toma en cuenta las porciones de calorías por cada proteína, carbohidratos y grasas.

- **Calcular Porciones de los Alimentos**

Para registrar un alimento en la dieta del cliente se desarrolló un método que permitirá sacar las porciones que el cliente consuma de ese alimento, tomando en cuenta las porciones generales con las que se registró el alimento por el administrador.

2.8 Tercera Iteración

La tercera iteración viene a comprender todo lo desarrollado para que el sistema recomiende automáticamente rutinas y dietas mediante el sistema experto.

2.8.1 Historia de usuario - Registrar Ejercicios y Alimentos

Para la historia de usuario del registro de ejercicio y alimentos con sus valores nutricionales, se realizó las siguientes acciones:

- **Creación de interfaz para administrar ejercicios y alimentos**

Se desarrolló en el sistema web del administrador una interfaz para que el administrador cree, actualice o elimine los datos para ejercicios y alimentos como se indica en la Figura30.

The figure consists of two screenshots of a web application interface. The top screenshot shows the 'Perfil' (Profile) page with a form for adding exercises. The form includes fields for 'Nombre' (Name), 'Musculo' (Muscle) with a dropdown, 'Información' (Information), 'Ejecución' (Execution), 'URL Video' (Video URL) with a placeholder 'www.youtube.com/embed/VIDEO_ID', and 'Imagen' (Image) with a file selection button. A 'Guardar' (Save) button is at the bottom. To the right is a 'VISTA PREVIA' (Preview) area. Below the form is a table with columns: '#', 'Nombre', 'Musculo', 'Información', 'Ejecución', 'Video', and 'Imagen'. The bottom screenshot shows the 'Dietas' (Diets) page with a form for adding food products. The form includes fields for 'Nombre (Descripción)' (Name (Description)), 'Cantidad (g)' (Quantity (g)), 'Calorías' (Calories), 'Proteínas (g)' (Proteins (g)), 'Carbohidratos (g)' (Carbohydrates (g)), and 'Grasas (g)' (Fats (g)). A 'Guardar' (Save) button is at the bottom. To the right is a large image showing a hand holding a calculator over a basket of vegetables.

Figura 30. Interfaces para la administración de ejercicios y alimentos

- **Creación de métodos para manejo de Ejercicios y Alimentos**

Se desarrolló los métodos para añadir, editar y eliminar dentro del servidor para manejo de datos de ejercicios y alimentos del sistema.

- **Relación con el sistema experto**

Los datos de ejercicios permitirán al sistema experto de rutinas presentar de una mejor forma los ejercicios con una imagen e información necesaria de cómo realizar el ejercicio. La relación de los alimentos con el sistema experto es mínima ya que está más orientada personalización de dietas por parte del cliente.

2.8.2 Historia de usuario - Actualizar Sistema Experto

Para la historia de usuario sobre la actualización del sistema experto de rutinas o dietas, se realizó las siguientes acciones:

- **Creación de Interfaz para actualizar sistema experto**

En el sistema Web del administrador se desarrolló dentro de las interfaces de ejercicios y dietas una pestaña en cada una en donde el administrador actualizara la base de conocimiento. En la Figura 31 se visualiza la pestaña sistema experto de cada interfaz.

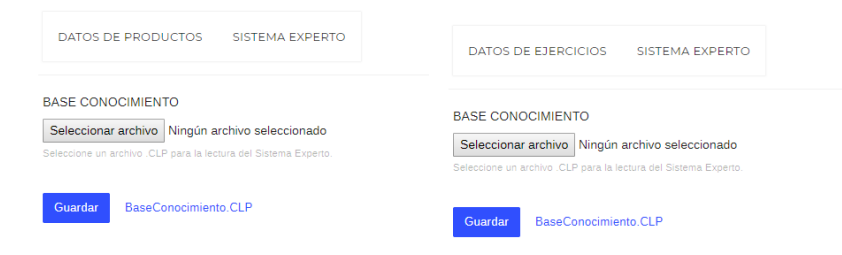


Figura 31. Interfaz de actualización del sistema experto

- **Método para envío y recepción de ficheros mediante servicios Web**

En el servidor se desarrolló un método para acceder al fichero que contiene toda la información de la base de conocimiento para el sistema experto y también se desarrolló un método para recibir un fichero y remplazar el original alojado en el framework de Spring mediante un servicio web.

2.8.3 Historia de usuario - Determinación de rutinas y dietas del Sistema Experto

Para la historia de usuario sobre la determinación de rutinas y dietas automáticas para cada cliente mediante el uso del sistema experto, se realizó las siguientes acciones:

- **Desarrollo de base de conocimiento de Rutinas y Dietas**

La base de conocimiento para determinar rutinas se desarrolló junto al usuario, ya que es el administrador del gimnasio y tiene conocimientos y experiencia de

acondicionamiento físico. Se tomo en cuenta los datos de los clientes como la edad, sexo, peso y altura para crear rutinas personalizadas evaluando estos campos.

En CLIPS para crear la base de conocimiento, se desarrolló un archivo .CLP, y mediante reglas creadas para cada caso, en la Figura 32 se presenta un ejemplo de la creación de una regla para generar una rutina en CLIPS:

```

1
2 (deftemplate diagnostico
3   (slot sexo)
4   (slot objetivo)
5   (slot talla)
6   (slot peso)
7 )
8 (deftemplate resultados_diagnostico
9   (slot ejercicio)
10  (slot series)
11  (slot repeticiones)
12 )
13 (defrule regla1
14   ?hecho1 <-(diagnostico (sexo varon))
15   ?hecho2 <-(diagnostico (objetivo abdomen))
16   ?hecho3 <-(diagnostico (talla ?t))
17   ?hecho4 <-(diagnostico (peso ?p))
18   (test (and (>= ?t 140) (<= ?t 180) ))
19   (test (and (>= ?p 30) (<= ?p 60) ))
20   =>
21   (retract ?hecho1 ?hecho2 ?hecho3 ?hecho4)
22   (assert (resultados_diagnostico (ejercicio elevacion_de_piernas) (series 3) (repeticiones 10)))
23   (assert (resultados_diagnostico (ejercicio inclinacion_lateral_con_mancuerna) (series 3) (repeticiones 12)))
24   (assert (resultados_diagnostico (ejercicio elevacion_de_rodillas_en_paralela) (series 3) (repeticiones 10)))
25 )
26

```

Figura 32. Creación de regla de rutinas en la base de conocimiento

Para la generación de la base de conocimiento de las dietas se tomó en cuenta los valores nutricionales del cliente, específicamente del campo de las calorías diarias necesarias que se calcula previamente con los datos del cliente. Mediante las calorías se desarrolló una dieta equilibrada que permita consumir el número de calorías necesarias del cliente. El desarrollo de dietas se lo realizó con un nutricionista que desarrollo 5 diferentes dietas por cada 100 calorías en un rango de 1500 y 3000 calorías. Igualmente, que las rutinas, se creó la base de conocimiento en CLIPS mediante reglas creadas para cada caso, en la Figura 33 se presenta un ejemplo para definir una dieta recomendada:

```

1 (deftemplate diagnostico
2   (slot calorías_necesarias)
3 )
4 (deftemplate resultados_diagnostico
5   (slot hora)
6   (slot descripción)
7   (slot calorías)
8   (slot proteínas)
9   (slot carbohidratos)
10  (slot grasas)
11 )
12 (defrule regla1
13   ?hecho3 <-(diagnostico (calorías_necesarias ?c))
14   (test (and (>= ?c 1500) (<= ?c 1600) ))
15   =>
16   (retract ?hecho1 ?hecho3 ?hecho4)
17   (assert (resultados_diagnostico (hora desayuno) (descripción una_taza_de_avena_cocida_con_una_vaso_de_leche_descremada) (calorías 25) (proteínas 5) (carbohidratos 5) (grasas 0)))
18   (assert (resultados_diagnostico (hora media_mañana) (descripción Una_pieza_de_fruta_fresca) (calorías 50) (proteínas 5) (carbohidratos 10) (grasas 0)))
19   (assert (resultados_diagnostico (hora almuerzo) (descripción Porción_de_pechuga_arroz_verduras_pan_tortilla) (calorías 550) (proteínas 50) (carbohidratos 100) (grasas 20)))
20   (assert (resultados_diagnostico (hora tarde) (descripción gelatina_galletas) (calorías 250) (proteínas 100) (carbohidratos 30) (grasas 5)))
21   (assert (resultados_diagnostico (hora cena) (descripción ensalada_con_atun_y_mostaza) (calorías 450) (proteínas 200) (carbohidratos 50) (grasas 10)))
22 )

```

Figura 33. Creación de regla de dietas en la base de conocimiento

- **Creación de servicios Web para evaluación de datos**

Se desarrolló servicios web en el servidor, para la comunicación con la base de conocimiento y evaluar los datos de un cliente presentando la respuesta del sistema experto en una lista que se enviara a la aplicación móvil para que el cliente del gimnasio visualice y guarde los datos si va a llevar a cabo las recomendaciones del sistema experto

2.9 Pruebas

Siguiendo el proceso de la metodología XP, en la fase de pruebas junto al administrador del gimnasio se realizaron pruebas de aceptación, las que según la metodología XP se deben realizar al final de cada iteración en base a las historias de usuario ya presentadas en la fase de planeación del proyecto.

2.9.1 Pruebas Primera Iteración

TABLA 24. PRUEBAS DE ACEPTACIÓN DE LA PRIMERA ITERACIÓN

CASO DE PRUEBA- HISTORIA DE USUARIO 1	
Nombre	Acceso al Portal Web
Descripción	El portal web estará disponible para el administrador desde cualquier navegador. Al cual podrá acceder mediante un usuario y una contraseña.
Condiciones de ejecución	El administrador debe estar registrado en el sistema
Entrada	El administrador entrará con la dirección web del sistema
	Insertará su usuario y contraseña
	El sistema se abrirá con todas las opciones que tiene el administrador
	Acceder a las diferentes interfaces del administrador
Resultado Esperado	Validación de datos
	No permitir el acceso a las interfaces sin registrarse primero
	Ingreso al sistema si los datos son correctos
Evaluación de la prueba	Satisfactoria
CASO DE PRUEBA- HISTORIA DE USUARIO 2	

Nombre	Registrar Clientes
Descripción	El administrador creará nuevos usuarios para el acceso a la aplicación móvil y también podrá dar de baja a los clientes ya registrados
Condiciones de ejecución	El administrador debe ingresar en el sistema
Entrada	El administrador entrará al sistema El administrador se dirigirá hacia la interfaz de clientes El administrador selecciona la opción de insertar El administrador ingresa la información del cliente y presiona el botón de guardar
Resultado Esperado	Validación de datos Nuevo registro creado en la base de datos y en el sistema reflejado el nuevo cliente Mensaje de alerta si los datos son incorrectos
Evaluación de la prueba	Satisfactoria
CASO DE PRUEBA- HISTORIA DE USUARIO 3	
Nombre	Administrar Pagos de clientes
Descripción	El administrador tendrá un registro de los pagos realizados por los clientes, los pagos pueden se los puede realizar en cuotas
Condiciones de ejecución	El administrador debe ingresar al sistema y debe existir registro de clientes previamente
Entrada	El administrador entrará al sistema El administrador se dirigirá hacia la interfaz de pagos El administrador ingresa el número de cedula de un cliente El administrador presiona el botón de pago nuevo El administrador selecciona una opción de pago Acepta los cambios y llena el campo de pago
Resultado Esperado	Validación de datos Nuevo pago creado para el cliente en la base de datos y en el sistema reflejado el nuevo pago Dependiendo del pago se registra la fecha en la que acaba la suscripción del cliente
Evaluación de la prueba	Satisfactoria

CASO DE PRUEBA- HISTORIA DE USUARIO 4	
Nombre	Notificar Fin Suscripción
Descripción	El sistema notificará por día, los clientes que ya finalizaron su suscripción o están próximos a finalizar, para recordar dar de baja a los clientes.
Condiciones de ejecución	El administrador ingresar en el sistema
Entrada	El administrador entra al sistema
	El administrador crea un pago que finaliza el mismo día de la prueba
	El administrador vuelve a ingresar al sistema
	En la parte superior derecha se mostrará que existe una notificación
	El administrador presionará el icono de la campana
	Se visualiza los usuarios que finalizaron suscripción
Resultado Esperado	Al entrar al sistema visualizar los usuarios que ya finalizaron la suscripción
	Al dar de baja al cliente que finalizo la suscripción, se elimina la notificación
Evaluación de la prueba	Satisfactoria
CASO DE PRUEBA- HISTORIA DE USUARIO 5	
Nombre	Promocionar ofertas
Descripción	El sistema presentará ofertas en la aplicación de los clientes para que ellos puedan visualizar
Condiciones de ejecución	El administrador debe ingresar en el sistema
Entrada	El administrador entrará al sistema
	El administrador se dirigirá hacia la interfaz de promociones
	El administrador ingresa una imagen de la oferta y crea un periodo de tiempo
	El cliente visualiza las ofertas creadas por el administrador en la aplicación móvil
Resultado Esperado	Validación de datos
	Guardar imágenes de ofertas en el servidor
	El cliente visualiza las ofertas dependiendo el periodo de tiempo

Evaluación de la prueba	Satisfactoria
CASO DE PRUEBA- HISTORIA DE USUARIO 6	
Nombre	Reportes de clientes
Descripción	El administrador podrá generar reportes de todos los clientes registrados y los clientes que estén dados de baja por el administrador
Condiciones de ejecución	El administrador debe ingresar en el sistema
Entrada	El administrador entrará al sistema
	El administrador se dirigirá hacia la interfaz de reportes
	El administrador selecciona una de las opciones para realizar el reporte
	El administrador imprime el reporte, o lo exporta a Excel o PDF
Resultado Esperado	Realizar reportes de clientes activos e inactivos
	Generar un PDF o Excel del reporte
	Imprimir el reporte
Evaluación de la prueba	Satisfactoria

2.9.2 Pruebas Segunda Iteración

TABLA 25. PRUEBAS DE ACEPTACIÓN DE LA SEGUNDA ITERACIÓN

CASO DE PRUEBA- HISTORIA DE USUARIO 7	
Nombre	Aplicación móvil
Descripción	Los clientes podrán acceder a una aplicación móvil del gimnasio mediante su correo electrónico y una contraseña.
Condiciones de ejecución	El cliente debe estar registrado por el administrador en el sistema
Entrada	El cliente abrirá la aplicación en su teléfono móvil

	El cliente ingresara su correo y contraseña y la guardara
	El cliente accede a las diferentes interfaces de la aplicación
Resultado Esperado	Validación de datos
	Permitir al cliente crear su contraseña
	Aplicación instalada en un dispositivo móvil
Evaluación de la prueba	Satisfactoria
CASO DE PRUEBA- HISTORIA DE USUARIO 8	
Nombre	Datos de clientes
Descripción	Los clientes del gimnasio registrarán una contraseña para acceso de la aplicación, además podrán actualizar sus datos personales y datos variables como el peso, altura, edad y medidas; para el funcionamiento del Sistema Experto
Condiciones de ejecución	El cliente tiene que ejecutar la aplicación móvil
Entrada	El cliente ingresara en la aplicación móvil
	Selecciona la segunda pestaña en la aplicación móvil
	Selecciona el botón de datos y medidas
	El cliente visualiza los datos personales
	El cliente selecciona el botón para editar y crea nuevos datos personales
Resultado Esperado	Validación de datos
	Interfaz atractiva para el cliente
	Se registran los datos personales en el sistema
Evaluación de la prueba	Satisfactoria
CASO DE PRUEBA- HISTORIA DE USUARIO 9	
Nombre	Generar valores nutricionales
Descripción	El cliente podrá visualizar sus valores nutricionales como IMC, peso ideal, calorías diarias necesarias, proteínas, carbohidratos y grasas recomendadas
Condiciones de ejecución	El cliente tiene que ejecutar la aplicación móvil, el cliente debe registrar los datos personales previamente
Entrada	El cliente ingresa en la aplicación móvil
	El cliente selecciona la segunda pestaña
	El cliente selecciona el botón de datos calculados

	El cliente visualiza los datos calculados que el sistema calculo con sus datos
Resultado Esperado	Interfaz atractiva para visualización de valores nutricionales Se crean automáticamente los valores nutricionales
Evaluación de la prueba	Satisfactoria
CASO DE PRUEBA- HISTORIA DE USUARIO 10	
Nombre	Pago pendiente de clientes
Descripción	El cliente podrá visualizar el último pago realizado y tendrá un registro de los pagos realizados.
Condiciones de ejecución	El cliente tiene que ejecutar la aplicación móvil, debe haber registro de pagos para el cliente
Entrada	El cliente ingresa a la aplicación móvil
	El cliente selecciona la segunda pestaña
	Selecciona el botón de pagos
	El cliente visualiza sus pagos
Resultado Esperado	Interfaz atractiva para la visualización de los pagos
	Visualizar el último pago que se registra en la plataforma web del administrador, directamente en el teléfono
Evaluación de la prueba	Satisfactoria
CASO DE PRUEBA- HISTORIA DE USUARIO 11	
Nombre	Registrar rutinas del cliente
Descripción	El cliente podrá generar su propia rutina diaria basándose en los ejercicios que estén registrados en el sistema. Además, tendrá un registro de las rutinas realizadas en los días anteriores.
Condiciones de ejecución	El cliente tiene que ejecutar la aplicación móvil, debe haber registro de ejercicios en el sistema
Entrada	El cliente ingresa a la aplicación móvil
	EL cliente a la sección de rutinas
	El cliente selecciona crear rutina
	El cliente añade ejercicios y llena los datos de repeticiones y series
	El cliente visualiza su rutina creada
Resultado Esperado	Validación de datos
	El cliente personaliza sus rutinas

	Se da una lista de ejercicios que puede realizar en el gimnasio
Evaluación de la prueba	Satisfactoria
CASO DE PRUEBA- HISTORIA DE USUARIO 12	
Nombre	Registrar dietas del cliente
Descripción	El cliente podrá generar su propia dieta diaria teniendo en cuenta las calorías, proteína, carbohidratos y grasas que el sistema le recomienda consumir; como ayuda el cliente se podrá basar en los alimentos y sus valores nutricionales que registre el administrador. Además, tendrá un registro de las dietas realizadas en los días anteriores.
Condiciones de ejecución	El cliente tiene que ejecutar la aplicación móvil, debe haber registro de ejercicios en el sistema
Entrada	El cliente ingresa a la aplicación móvil
	El cliente va a la sección de dietas
	El cliente selecciona crear dieta
	El cliente presiona el botón de añadir dieta
	El cliente llena los datos de la dieta o añade alimentos de una lista creada por el administrador
	El cliente visualiza los valores nutricionales que contiene su dieta
Resultado Esperado	Validación de datos
	El cliente personaliza sus dietas
	El cliente puede calcular los valores nutricionales con los alimentos ingresados por el administrador
Evaluación de la prueba	Satisfactoria

2.9.3 Pruebas Tercera Iteración

TABLA 26. PRUEBAS DE ACEPTACIÓN DE LA TERCERA ITERACIÓN

CASO DE PRUEBA- HISTORIA DE USUARIO 13	
Nombre	Registrar ejercicios y alimentos
Descripción	El administrador creará los ejercicios y alimentos con sus valores nutricionales, para ser usados como complemento del Sistema Experto

Condiciones de ejecución	El administrador ingresa en el sistema
Entrada	El administrador ingresa al sistema web El administrador ingresa a la pantalla de rutinas El administrador visualiza y crea nuevos ejercicios El administrador ingresa a la pantalla de dietas El administrador visualiza y crea nuevos valores de alimentos El cliente inicia la aplicación y procede a crear alimentos o dietas
Resultado Esperado	Validación de datos Registrar en el sistema ejercicios que puede realizar el cliente en el gimnasio Registrar en el sistema alimentos con valores nutricionales Visualizar los valores nutricionales de los alimentos creados, en la aplicación móvil Visualizar los ejercicios creados en la aplicación móvil
Evaluación de la prueba	Satisfactoria
CASO DE PRUEBA- HISTORIA DE USUARIO 14	
Nombre	Actualizar sistema experto
Descripción	El administrador podrá descargar la base de conocimiento para poder realizar actualizaciones y subir nuevamente al servidor
Condiciones de ejecución	El administrador ingresa al sistema
Entrada	El administrador ingresa al sistema web El administrador ingresa a la pantalla de rutinas El administrador selecciona sistema experto El administrador descarga o sube un nuevo documento. clips de la base de conocimiento de rutinas El administrador ingresa a la pantalla de dietas Realiza el mismo procedimiento de rutinas
Resultado Esperado	Descargar la base de conocimiento Subir al servidor una nueva versión de la base de conocimiento
Evaluación de la prueba	Satisfactoria
CASO DE PRUEBA- HISTORIA DE USUARIO 15	
Nombre	Determinación de rutinas y dietas del sistema experto

Descripción	Los clientes podrán seleccionar que tipo de ejercicio desean realizar para que el sistema genere una rutina tomando en cuenta sus datos ingresados. De la misma manera la dieta deberá generarse automáticamente, tomando en cuenta los valores nutricionales
Condiciones de ejecución	Debe existir estar registrado el cliente
Entrada	El cliente ingresa en la aplicación móvil
	El cliente se dirige hacia rutinas
	El cliente selecciona rutina recomendada
	El cliente selecciona un tipo de ejercicio
	El cliente guarda la rutina recomendada
	El cliente se dirige hacia dietas
	El cliente selecciona dieta recomendada
	El cliente guarda la dieta recomendada
Resultado Esperado	Interfaz atractiva para visualización de rutina y dietas
	Se visualiza los ejercicios recomendados con sus detalles
	Se visualiza las diferentes comidas al día con sus calorías
	Se crea automáticamente una rutina o dieta dependiendo los datos del cliente
Evaluación de la prueba	Satisfactoria

2.9.4 Resultado de pruebas

Las pruebas realizadas para el Sistema experto web móvil para determinación de rutinas y dietas, buscan validar que todos los casos de usos planteados con sus respectivos requisitos vinculados se cumplan en totalidad y de manera correcta, para este sistema se decidió realizar las pruebas de usuario por cada historia de usuario y agruparlas por cada iteración.

En la Tabla 27 se refleja los requisitos funcionales que se los detalla con el código RF y casos de uso involucrados con código CU, de cada historia de usuario de prueba presentada con la sigla H.

TABLA 27. REQUISITOS Y CASOS INVOLUCRADOS DE PRUEBAS

		PRIMERA ITERACIÓN						SEGUNDA ITERACIÓN						TERCERA ITERACIÓN		
		H01	H02	H03	H04	H05	H06	H07	H08	H09	H10	H11	H12	H13	H14	H15
REQUISITOS FUNCIONALES	RF-01	X	X	X	X	X	X							X	X	X
	RF-02	X	X				X		X							
	RF-03	X		X	X						X					
	RF-04	X												X		
	RF-05	X													X	
	RF-06							X	X	X	X	X	X			
	RF-07							X		X						
	RF-08							X				X				X
	RF-09							X					X			X
	RF-10					X										
CASOS DE USO	CU-01	X	X	X	X	X	X							X	X	X
	CU-02	X														
	CU-03	X	X													
	CU-04	X		X	X											
	CU-05	X													X	
	CU-06	X												X		
	CU-07	X													X	
	CU-08	X												X		
	CU-09	X					X									
	CU-10	X				X										
	CU-11							X	X	X	X	X	X			
	CU-12							X	X							
	CU-13							X			X					
	CU-14							X				X				
	CU-15							X								X
	CU-16							X					X			
	CU-17							X								X
	CU-18							X		X						
	CU-19							X								
	CU-20					X		X								

Los resultados por cada iteración se evaluaron tomando en cuenta los requisitos y casos de uso que se consumaban por cada prueba.

TABLA 28. RESULTADOS PRIMERA ITERACIÓN

Nombre Prueba	Acceso al Portal Web	Registrar Clientes	Administrar pago de clientes	Notificar Fin Suscripción	Promocionar ofertas	Reporte Clientes
Requisitos Involucrados	5	2	2	2	2	2
Requisitos Probados	5	2	2	2	2	2
Casos de Uso Involucrados	10	2	2	2	3	2
Casos de uso Probados	10	2	2	2	3	2
Resultado	Éxito	Éxito	Éxito	Éxito	Éxito	Éxito

TABLA 29. RESULTADOS SEGUNDA ITERACIÓN

Nombre Prueba	Aplicación móvil	Datos de Clientes	Generar valores nutricionales	Pago pendiente del cliente	Registrar Rutinas del Cliente	Registrar Dietas del Cliente
Requisitos Involucrados	4	2	2	2	2	2
Requisitos Probados	4	2	2	2	2	2
Casos de Uso Involucrados	10	2	2	2	2	2
Casos de uso Probados	10	2	2	2	2	2
Resultado	Éxito	Éxito	Éxito	Éxito	Éxito	Éxito

TABLA 30. RESULTADOS TERCERA ITERACIÓN

Nombre Prueba	Registrar ejercicios y alimentos	Actualizar Sistema Experto	Determinación de rutinas y dietas del Sistema Experto
Requisitos Involucrados	2	2	3
Requisitos Probados	2	2	3
Casos de Uso Involucrados	3	3	3
Casos de uso Probados	3	3	3
Resultado	Éxito	Éxito	Éxito

En la Figura 34 se presenta en un diagrama de barras los requisitos y casos de usos cumplidos por cada iteración.

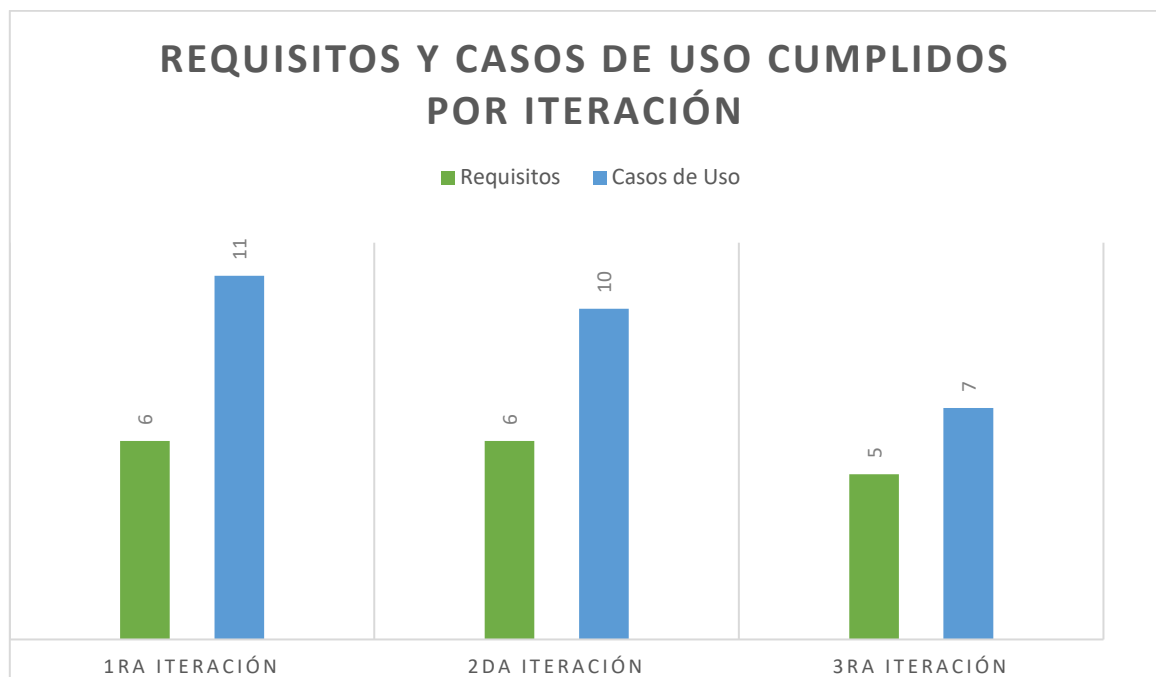


Figura 34. Requisitos y casos de uso cumplidos

CAPITULO III

IMPLEMENTACIÓN

3.1 Instalación

El sistema tiene tres diferentes plataformas para la instalación:

- **Servidor Spring:** Al administrador se le entrega un ejecutable que deberá ser montado en un servidor Web que trabaje con el lenguaje de programación java para el funcionamiento de los servicios Rest.
- **Servidor Angular:** El portal web del administrador esta desarrollado para que una persona con una base de conocimientos en sitios web pueda realizar la instalación, la instalación no es necesariamente que sea en un servidor web ya que puede ser instalado en una maquina personal debido que no consume muchos recursos de hardware o software como el servidor de Spring.
- **Aplicación móvil:** Por parte de la aplicación móvil, se entrega un ejecutable para teléfonos Android el cual puede ser instalado fácilmente por cualquier.

3.2 Requisitos

Los requisitos para usar el sistema son los siguientes:

Servicios Web - Spring Framework

- Acceso a Internet
- Servidor Web

Portal Web del Administrador – Angular

- Acceso a internet
- Navegador web
- Servidor Web

Aplicación del Cliente - Ionic

- Acceso a Internet
- Dispositivo móvil (Android 6 o superior)

3.2.1 Requisitos del servidor para despliegue de servicios Web

Al momento de escoger un servidor para desplegar los servicios web se debe tomar en cuenta los requisitos presentados en la Tabla 31 para que los servicios funcionen correctamente.

TABLA 31. REQUISITOS DEL SERVIDOR DE SERVICIOS WEB

Lenguaje de programación	Java
SGBD	PostgreSQL
Espacio de disco duro	10gb o superior
Memoria RAM	1GB o superior
Disponibilidad	Alta

Normalmente se busca un servidor con Spring Boot, pero lo más recomendable es instalarlo en un servidor Tomcat, y realizar las configuraciones necesarias para conectar la base de datos con el sistema. Se recomienda estos dos servidores por que las pruebas se las realizaron con estos servidores.

3.2.2 Requisitos del servidor de la plataforma web del administrador

La plataforma web del administrador puede ser ejecutada directamente desde una computadora con pocos requerimientos y con acceso a internet; pero para este caso se tomará en cuenta los requisitos de alojamiento en un servidor web como se indica en la Tabla 32:

TABLA 32. REQUISITOS DEL SERVIDOR PARA LA PLATAFORMA WEB

Lenguaje de programación	HTML5, CSS3, JavaScript
Espacio de disco duro	500mb o superior
Memoria RAM	1GB o superior
Otros	Acceso a Internet

3.3 Alojamiento

Debido a que son dos diferentes frameworks que necesitan estar alojados en un servidor para su ejecución, el alojamiento se lo realiza de distinta manera para cada uno.

3.3.1 Alojamiento de Spring Framework

Spring es un framework del lenguaje de programación java por lo tanto tiene similitud con proyectos desarrollados en el mismo lenguaje, por lo que al momento de compilación lo único que se obtiene es un archivo (.war) que no es más que un archivo comprimido que se utiliza para el manejo de aplicaciones web. Para alojarlo en el servidor solo se necesita subirlo al servidor de aplicación java y ejecutarlo, el servidor lo reconocerá como un proyecto java y lo pondrá en producción.

Es necesario antes de compilar el proyecto java, tener la dirección de la base de datos y las credenciales de acceso, ya que se debe configurar primero la conexión con la base de datos para que funcione correctamente los servicios web.

3.3.2 Alojamiento de Angular Framework

La estructura del proyecto desarrollado en angular integra en un directorio todos los archivos y componentes necesarios para su ejecución. Se tiene que configurar la dirección de los servicios web en la aplicación antes de alojarla en el servidor

3.4 Despliegue

El despliegue del sistema desarrollado en Angular y Spring se lo realiza después de alojar todo lo necesario de cada framework en el servidor y ejecutando el servidor para realizar las diferentes pruebas en una diferente máquina para probar el acceso al sistema.

El despliegue de la aplicación móvil consta únicamente en desarrollar el archivo para la instalación en los dispositivos móviles teniendo en cuenta cambiar los dominios de los servicios web para su conexión con el servidor principal.

Para validar el correcto funcionamiento del sistema web y la aplicación móvil se lo realizó evaluando el porcentaje de satisfacción que se obtuvo de las pruebas de aceptación que se desarrollaron junto el usuario administrador además se tomó en cuenta las observaciones que se dio de cada prueba como se indica en la Tabla 33.

TABLA 33.RESULTADOS DE PRUEBAS DE ACEPTACIÓN

Historia de Usuario	Descripción	Porcentaje de Satisfacción	Observaciones
01. Acceso al Portal Web	El portal web estará disponible para el administrador desde cualquier navegador. Al cual podrá acceder mediante un usuario y una contraseña	100%	Con la prueba se determinó que faltaba mensajes de alerta al iniciar sesión por lo que se procedió a implementarlas
02. Registrar Clientes	El administrador creará nuevos usuarios para el acceso a la aplicación móvil y también podrá dar de baja a los clientes ya registrados	100%	Se obtuvo un problema con los campos ingresados de los clientes y se procedió a realizar validaciones usando html5
03. Administrar pago de clientes	El administrador tendrá un registro de los pagos realizados por los clientes, los pagos pueden se los puede realizar en cuotas	100%	Los pagos de clientes generaban muchos datos, por lo que se realizó cambios en las tablas de pagos para mejorar su visualización
04. Notificar Fin Suscripción	El sistema notificará por día, los clientes que ya finalizaron su suscripción o están próximos a finalizar, para recordar dar de baja a los clientes	100%	Al momento de realizar el nuevo pago del cliente que finalizó la suscripción, la notificación seguía activa. Por lo que se procedió a realizar dichos cambios.

05. Promocionar ofertas	El sistema presentará ofertas en la aplicación de los clientes para que ellos puedan visualizar.	100%	Se ejecutó sin problema la prueba.
06. Reporte Clientes	El administrador podrá generar reportes de todos los clientes registrados y los clientes que estén dados de baja por el administrador	100%	Al crear un archivo Excel con los reportes, el archivo aparece como dañado. Al administrador se le recomendó que ejecute la herramienta de recuperación de Excel para armar la tabla correctamente
07. Aplicación móvil	Los clientes podrán acceder a una aplicación móvil del gimnasio mediante su correo electrónico y una contraseña.	100%	Error al iniciar sesión de algunos clientes. El archivo o imagen era muy pesado por lo que produjo errores al cargar los datos
08. Datos de Clientes	Los clientes del gimnasio registrarán una contraseña para acceso de la aplicación, además podrán actualizar sus datos personales y datos variables como el peso, altura, edad y medidas; para el funcionamiento del Sistema Experto	100%	La edad y el sexo del cliente debía ser ingresado a cada momento. Mediante esta prueba se optimizó el proceso obteniendo los datos desde el administrador

09. Generar valores nutricionales	El cliente podrá visualizar sus valores nutricionales como IMC, peso ideal, calorías diarias necesarias, proteínas, carbohidratos y grasas recomendadas	100%	El administrador recomendó que se visualice el peso ideal de cada persona
10. Pago pendiente del cliente	El cliente podrá visualizar el último pago realizado y tendrá un registro de los pagos realizados	100%	Se obtuvo los resultados esperados
11. Registrar Rutinas del Cliente	El cliente podrá generar su propia rutina diaria basándose en los ejercicios que estén registrados en el sistema. Además, tendrá un registro de las rutinas realizadas en los días anteriores	100%	En las pruebas preliminares se notó que los valores de comidas ingresados por el administrador debían ser divididos por cada gramo a utilizar. Además, el administrador recomendó utilizar gramos diferentes para cada proteína carbohidrato o grasa
12. Registrar Dietas del Cliente	El cliente podrá generar su propia dieta diaria teniendo en cuenta las calorías, proteína, carbohidratos y grasas que el sistema le recomienda consumir; como ayuda el cliente se podrá basar en los alimentos y sus valores nutricionales que registre el administrador.	100%	Ninguna

	Además, tendrá un registro de las dietas realizadas en los días anteriores		
13. Registrar Ejercicios y Alimentos	El administrador creará los ejercicios y alimentos con sus valores nutricionales, para ser usados como complemento del Sistema Experto	100%	Se generó la prueba con los resultados esperados
14. Actualizar Sistema Experto	El administrador podrá descargar la base de conocimiento para poder realizar actualizaciones y subir nuevamente al servidor	80%	Se obtuvo los resultados esperados, pero se recomendó usar como referencia el fichero de descarga para crear nuevas reglas
15. Determinación de rutinas y dietas del Sistema Experto	Los clientes podrán seleccionar que tipo de ejercicio desean realizar para que el sistema genere una rutina tomando en cuenta sus datos ingresados. De la misma manera la dieta deberá generarse automáticamente, tomando en cuenta los valores nutricionales	100%	Se necesitó realizar varias pruebas con el administrador para dejar correctamente configurado el sistema experto

CONCLUSIONES

1. Las herramientas de desarrollo propuestas en este proyecto permitieron que la programación sea más sencilla ahorrando el esfuerzo en la codificación para las configuraciones con la base de datos o incluso permitiendo su integración.
2. La implementación del sistema administrador permitió obtener un registro de todos los clientes, además de un estricto control del cumplimiento de pagos de los servicios que ofrece el del gimnasio.
3. El desarrollo de la aplicación móvil para el gimnasio optimizó el trabajo de los entrenadores para crear una rutina de ejercicios y dieta personalizada con sus valores nutricionales logrando cumplir los objetivos propuestos.
4. La implementación de la metodología XP permitió que el proyecto se encuentre estructurado agilizando el proceso de desarrollo y calidad del software.

RECOMENDACIONES

1. Al momento de escoger las herramientas de desarrollo hay que tener en cuenta si éstas son compatibles entre sí y exista la información necesaria para su integración y todo lo que se va a requerir en el desarrollo del sistema.
2. Buscar información de sistemas expertos en libros de inteligencia artificial, ya que comúnmente se toma como referencia estos sistemas para explicar algunas de las aplicaciones.
3. Establecer un formato de los datos a utilizar en el sistema experto que permitirá evaluarlos, clasificarlos y estructurarlos a fin de crear la base de conocimientos.
4. Evitar actualizar los frameworks mientras se encuentra en la etapa de producción porque pueden traer inconvenientes al momento de ejecutar.
5. Realizar más investigaciones y sistemas en el área de inteligencia artificial que permitan reforzar conocimientos adquiridos en el transcurso de la carrera.

REFERENCIAS

- Badaró, S., Ibañez, L. J., & Agüero, M. J. (2013). Sistemas Expertos Fundamentos, Metodologías y Aplicaciones. DIALNET.
- Bahit, E. (2012). Scrum y eXtreme Programming para Programadores. Autoedición.
- Benítez, R., Escudero, G., Kanaan, S., & Rodó, D. M. (2014). Inteligencia Artificial Avanzada. Barcelona: Editorial UOC.
- Cadena, C., Quimí, R. D., & Peláez, E. (2015). Análisis, diseño y desarrollo de un sistema experto con tecnología web. ESPOL.
- Canós, J., Letelier, P., & Carmen, P. (2012). Metodologías Ágiles en el Desarrollo de Doftware. Universidad Politécnica de Valencia.
- Cordero Morales, D., Ruiz Constanten, Y., & Torres, Y. (2013). Sistema de Razonamiento Basado en Casos para la identificación de riesgos de software. Revista Cubana de Ciencias Informáticas.
- Deitel, P., Deitel, H., & Deitel, A. (2014). Internet & World Wide Web Cómo Programar. Mexico: PEARSON EDUCACIÓN.
- Dorta, M. R. (2016). AngularJS paso a paso. LeanPub.
- Ginestà, M. G., & Mora, O. P. (2012). Bases de datos en PostgreSQL. Cataluña: Universidad Abierta de Cataluña.
- Goins, A. (2017). Ionic Framework: Construa aplicativos para todas as plataformas mobile. Brasil: Casa do Código.
- Joseph C. Giarratano, P. (2015). CLIPS Reference Manual. Obtenido de www.clipsrules.net
- Laínez Fuentes, J. R. (2015). Desarrollo de Software Ágil: Extremme Programming y Scrum. 2ª Edición. IT Campus Academy.
- Nilsson, N. J. (2014). Principles of Artificial Intelligence. Morgan Kaufmann.
- Point, T. (2018). IONIC. Tutorials Point.
- Regina, O., & Leo, H. (2015). PostgreSQL: Up and Running. Sebastopol, CA: O'Reilly Media, Inc.
- Rivadeneira Molina, S. G. (2013). Metodologías ágiles enfocadas al modelado de requerimientos. Universidad Nacional de la Patagonia Austral.
- Romero, R., & Elena, M. (2015). ANÁLISIS, DISEÑO Y DESARROLLO DE UN BALANCED. UNIVERSIDAD CENTRAL DEL ECUADOR.
- Serrano, A. G. (2013). Inteligencia Artificial: Fundamentos, práctica y aplicaciones. Madrid: Alfaomega.
- Solis, C. (2015). Manual del Guerrero AngularJS.
- Treviño, J. A. (2012). ESTIMACIÓN DEL GASTO ENERGÉTICO PARA EL ADULTO. IV Simposio de nutrición clínica.
- Walls, C. (2015). Spring Cuarta Edición. Denver, CO: ANAYA MULTIMEDIA.

ANEXOS

- **ANEXO 1:** MANUAL TÉCNICO (EN CD)